Cross-Database Facial Expression Recognition Based on Fine-Tuned Deep Convolutional Network

Marcus Vinicius Zavarez*, Rodrigo F. Berriel and Thiago Oliveira-Santos
Universidade Federal do Espirito Santo, Brazil
Email: *vini.vni@gmail.com

***Abstract*—Facial expression recognition is a very important research field to understand human emotions. Many facial expression recognition systems have been proposed in the literature over the years. Some of these methods use neural network approaches with deep architectures to address the problem. Although it seems that the facial expression recognition problem has been solved, there is a large difference between the results achieved using the same database to train and test the network and the cross-database protocol. In this paper, we extensively investigate the performance influence of fine-tuning with cross-database approach. In order to perform the study, the VGG-Face Deep Convolutional Network model (pre-trained for face recognition) was fine-tuned to recognize facial expressions considering different well-established databases in the literature: CK+, JAFFE, MMI, RaFD, KDEF, BU3DFE, and AR Face. The cross-database experiments were organized so that one of the databases was separated as test set and the others as training, and each experiment was ran multiple times to ensure the results. Our results show a significant improvement on the use of pre-trained models against randomly initialized Convolutional Neural Networks on the facial expression recognition problem, for example achieving 88.58%, 67.03%, 85.97%, and 72.55% average accuracy testing in the CK+, MMI, RaFD, and KDEF, respectively. Additionally, in absolute terms, the results show an improvement in the literature for cross-database facial expression recognition with the use of pre-trained models.**

I. Introduction

Facial expression recognition is a very important research field to understand human emotions. The human brain can recognize facial expressions only by the face characteristics. Although recognition of facial expressions seems to be a simple task for humans, it is quite difficult to be performed by computers.

In the facial expression recognition (FER) problem, there are six basic universal [2] expressions that are recognized in several different cultures and are widely used in the literature: fear, sad, angry, disgust, surprise and happy. Some works in the literature also take the neutral expression in consideration, which sums up to the seven expressions also widely used in the literature. Many facial expression recognition systems have been proposed in the literature over the years [3]–[9]. Although it seems that the facial expression recognition problem has been solved, there is a large difference between the results achieved using the intra and the cross-database protocols. In the intra-database protocol (i.e., training in one database and testing in a subject-independent set of the same database), the current methods already achieve high accuracies, reaching around 95% [3], [4], [6]. In the other hand, methods evaluated using the cross-database protocol (i.e., training in one or more databases and evaluating in different databases) do not report high accuracies, ranging between 40% and 66% [3]–[9]. In this context, the evaluation of a method using the intra-database protocol seems to provide limited insight into the generalization capability of such method.

Some of the methods employed nowadays for facial expression recognition [6], [8], [10] use neural network approaches with deep architectures to address the problem. One specific type of deep network is the Convolutional Neural Networks (CNNs) proposed by [11]. A usual constraint of CNNs is the need of big amounts of data to ensure the convergence of their training algorithms ir order to achieve good accuracies. However, not all applications have the necessary amount of data to train these models, either due to costs of data acquisition or other application constraints. To deal with this limitation, some works apply fine-tuning techniques to transfer learning from one problem to the other. In these cases, instead of randomly initializing the weights of a CNN, these procedures use the weights of a CNN that has been previously trained with an extensive set of images to speed up the convergence of the training algorithm. Some methods [12]–[14] employed fine-tuning techniques for the facial expression recognition problem with intra-database protocol.

Nevertheless, facial expression recognition methods (even those using deep neural networks) struggle to achieve high accuracies when evaluated using the cross-database protocol. Moreover, most of the related works (especially those using cross-database protocol) do not perform an extensive experimentation (usually training in a single database and testing in another one, i.e., using few databases). This also limits the evaluation of the generalization of these methods. In addition, there are many publicly available databases in the literature, but there is no consensus for the cross-database protocol evaluation. Some of these works use databases that are not freely available. Besides that, it is even hard to ensure that the very same database is being used in the same manner. As some databases are converted from video files, different authors end up using different techniques to extract the images of interest, which may mislead the comparisons. Given these facts, there is still need for more extensive investigation using the cross-database protocol. Even though the environment is controlled within database (frontal face images, same ethnicity subjects, similar light conditions, no occurrence of occlusion, among others), it is not controlled across databases.

In this paper, we propose an extensive experimentation to evaluate the performance of a fine-tuned deep neural network in the facial expression recognition problem using the cross-database protocol. To perform this study, we fine-tuned the VGG network pre-trained in a face recognition database (VGG-Face, as originally proposed) to recognize facial expressions considering different well-established databases in the literature: AR Face Database [15], Extended Cohn-Kanade Database (CK+) [16], Binghamton University 3D Facial Expression (BU3DFE) [17], The Japanese Female Facial Expression (JAFFE) [18], MMI [19], Radboud Faces Database (RaFD) [20] and Karolinska Directed Emotional Faces (KDEF) [21] databases. These datasets altogether comprise more than 6,200 images from subjects of different ethnicities, genders, and ages in a variety of environments and in both spontaneous and posed facial expressions. The experiments were organized so that one of the databases was separated as test set and the others as training sets (i.e. leave-one-out). In addition, the experiments were ran multiple times to account for the randomness of the proposed method and to allow for a more robust analysis of the results. Indeed, results showed variation among different seeds (e.g. the accuracy of the 10 runs of the VGG-Random with MMI varied from 46.67% to 59.74%), which indicates the need for multiple runs when evaluating Deep Neural Networks. When comparing fine-tuned models with randomly initialized ones, results showed that, in general, fine-tuned models perform better. However, there are some cases where randomly initialized models performed better (e.g. evaluating in the JAFFE database) than fine-tuned ones. Finally, in absolute terms, our models achieved state-of-art results for most of the databases: 88.58%, 67.03%, 85.97%, and 72.55% of average accuracy (CK+, MMI, RaFD, and KDEF, respectively). The only case our models did not outperform the literature was for the JAFFE database (44.32%), which is a highly biased database in terms of gender and ethnicity, and also yields the worst results in the literature.

II. Related Works

There are several methods proposed to address the facial recognition problem in the literature. Many of them focus evaluation within the same database and therefore can only prove their effectiveness within the same conditions of the training database (e.g. [6], [22], [23]). Among these works, there are still some that do not ensure a separation of subjects in the training and test sets, and therefore they cannot even ensure effectiveness within the same database. Instead, they give a false impression of high accuracy (e.g. [24]–[27]). In this section, focus is given to related works performing cross-database evaluations. It is important to note that some of the works in the literature exclude the neutral expression from the problem, but these are not the focus of this study.

Shan et al. [3] proposed a support vector machine (SVM) combined with other machine learning methods, and evaluated their method using both intra and cross-database protocols. The authors used CK+, MMI, and JAFFE databases in a 10-fold cross-validation using the intra-database protocol. Their method achieved an average of 91.4%, 86.9%, and 81.0% of accuracy, respectively. Moreover, they performed cross-database experiments training with CK+ database and testing in the MMI and JAFFE databases. Their method achieved an accuracy of 51.1% and 41.3%, respectively. Zhang et al. [4] use a multiclass support vector machine (SVM) based on a multiple kernel learning (MKL) to perform facial expression recognition and they evaluated their system with the CK+ and MMI databases. Using the intra-database protocol, they achieved 93.6% and 92.8% of accuracy, respectively. For cross-database experiments, when training with MMI database and testing in the CK+ database, their system achieved 61.2% of accuracy. In addition, their method achieved 66.9% of accuracy when training with CK+ database and testing in the MMI. This indicates that intra-database experiments are easier than cross-database, and they cannot be used to generalize the performance of a method.

Lopes et al. [6] proposed a method that uses a combination of image pre-processing and Convolutional Neural Networks. These pre-processing steps include spatial normalization, image cropping, and intensity normalization, and help to extract specific features for expression recognition. The authors also employed a data augmentation procedure that included the generation of synthetic samples to cope with the lack of data. Cross-database experiments were also performed, but they were not the focus of their work. They trained their model only with CK+ and evaluated on the JAFFE and BU3DFE databases, achieving an accuracy of 37.36% and 42.25%, respectively. A Boosted Deep Belief Network (BDBN) was proposed by [8]. Their method proposed a composition of weak binary classifiers having each of them responsible for classifying one expression. The BDBN combines feature learning, feature selection, and classifier construction in a unified framework. A cross-database approach was used to evaluate the generalization of their method, i.e. their BDBN was trained with the CK+ database and tested in the JAFFE database, achieving a performance rate of 68.00%. However, this performance rate is not directly comparable with the other methods in the literature, because they used a composition of binary accuracies to derive this metric. Mayer et al. [23] uses a support vector machine to classify the expressions, also used cross-database protocol in their evaluation. Their experiments comprises one-versus-one comparisons of the CK, MMI, and FEEDTUM databases (i.e. only one database used to train and only another one to test). When training with MMI and testing in the CK+, their system achieved 60.8%; and when training with CK+ for training and testing in the MMI their system achieved 53.2% of accuracy.

In recent studies, [7] proposed a Convolutional Neural Network that consists of two convolutional layers followed by max pooling and four Inception modules. The authors used seven standard databases (Multi-PIE, MMI, CK+, DISFA, FERA, SFEW and FER2013) to perform a leave-one-out experiment having each of the databases as testing set. Their method achieved an accuracy of 64.2% when testing in the

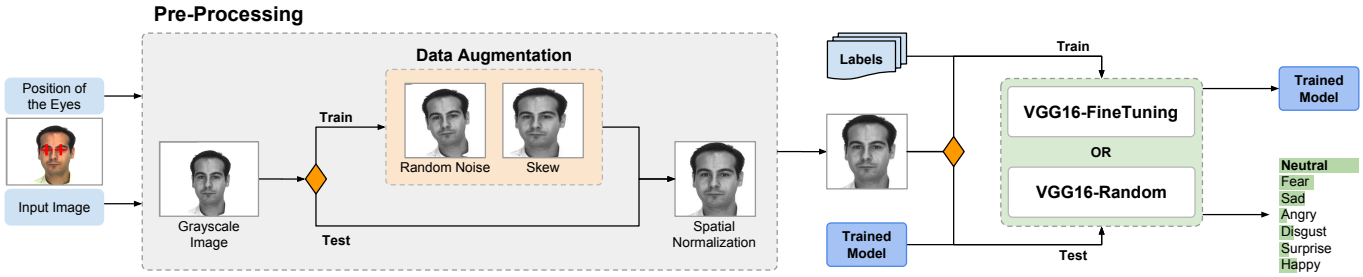

Fig. 1. The proposed method starts with pre-processing to generate the input images for the CNN. Therefore, the original image is firstly transformed into gray scale. Subsequently, a data augmentation step is performed to increase the number of images in the database (this step is only performed during training). Later, a spatial normalization is performed to correct major rotation, translation, and scale problems using the position of the center of the eyes. Finally, the CNN can be trained to generate a model and be later used to predict the facial expression of a given image.

CK+ database and training with the other six databases, and 55.6% when testing in the MMI database and training with other databases. It is important to note that the Multi-PIE database is not publicly available, therefore it is hard to reproduce their results. Hasani et al. [28] proposed a spatio-temporal two-parts network that uses a deep neural network and a conditional random fields (CRF) module to recognize facial expressions in sequence of images (i.e. videos). The DNN-based network contains three InceptionResNet modules and two fully-connected layers that capture spatial relations of facial expression on images. The CRF module captures the temporal relation between the frames. The authors use CK+, MMI, and FERA databases with leave-one-out approach. The method reported an accuracy of 73.91% and 68.51% to the CK+ and MMI databases respectively. However, considering single-frame evaluation (i.e. without the CRF module) such as approached in this work, their system reported 64.81% and 52.83% of accuracy for the CK+ and MMI, respectively.

As can be seen, most of the works in the literature do not provide a way to reproduce their results. Moreover, for the video databases, it is even hard to ensure the same set of images are used across different works. In addition, many of these works only employ the cross-database protocol using only one database during training and only one for test. Regardless the differences in the evaluation protocols, the cross-database accuracies are still very low when compared to the intra-database. In this context, there is still need for a more extensive experimentation, using multiple databases, and allowing for reproducibility.

III. CNN Facial Expression Recognition System

The proposed facial expression recognition system comprises two main modules (Figure 1): a pre-processing step and the Convolutional Neural Network. The first module is responsible for preparing the input image, augmenting the samples, and performing a spatial normalization. The second module is responsible for performing the training of the model and later classifying the image in one of the seven allowed expressions. The system receives an image of a face with the position of the center of each eye as input, and outputs the expression with the highest confidence.

A. Pre-processing

The pre-processing begins with the conversion of the input image to grayscale. This step is performed to minimize the variation of the images between the databases, given that some of them are already in grayscale. As the Convolutional Neural Network (ConvNet) described later expects a 3-channel input image, this grayscale image is replicated in the three channels. Subsequently, an offline data augmentation step is performed to increase the number of images in the database. The number of samples generated varies according to each combination of databases used in the training phase. Later, a spatial normalization is performed to correct major rotation, translation, and scale problems using the position of the center of the eyes. This step is also performed for all the images of every database (both training and test sets). The following subsection describes each of these steps in details.

1) Offline Data Augmentation: ConvNets require large sets of data in order to be able to generalize to a given problem. However, publicly available databases for facial expression recognition do not have enough images to address this problem. Simard et al. [29] proposed a data augmentation technique to increase the database through generation of synthetic samples for each original image. Inspired in this technique, the following operations were applied offline as data augmentation: a random noise was added to the position of the eyes and skew. For each image in the original database, 10 synthetic images were generated, where a random noise was added to the position of the eyes on 70% of the synthetic images and a random skew was applied on the remaining 30%.

The rotation, translation, and scale procedures consist in adding a random noise in the position of the eyes before performing the spatial normalization. Therefore, the spatial normalization with the random noise added to the position of the eyes is equivalent to performing a small rotation, translation, and/or scaling. The noise is randomly generated by sampling from a Gaussian distribution with standard deviation equals to 10% of the distance between the eyes.

The skew procedure consists of changing the corners of the image to generate a distortion. Firstly, the side of the image (left, right, top, bottom) in which the skew will be applied is randomly chosen. Secondly, the amount of skew

applied to the image is selected by sampling from a uniform distribution varying between 2% and 15% of the length of the side. Finally, the two corner points are changed to generate a distorted image. It is important to note that the skew operation is performed after the correction of rotation, translation, and scale described in the next section.

Figure 3 shows the distribution of all expressions of the databases used in this study. As it can be seen, some expressions have more samples than the others. This difference between the amount of samples of each expression can make the training of the network to give more weight to the expression with more samples. To minimize the problem of having a biased trained model, the databases of each training configuration were balanced before training. Therefore, the appropriate number of extra synthetic images was generated for each expression in each database. This extra number of synthetic images is referred as complement. The expressions were balanced proportionally according to the one with the most samples considering all databases as in Equation 1:

$$C_e^d = \left(\frac{M - T_e}{T_e} \right) \cdot n_e^d \tag{1}$$

$$M = max(T_e), \forall \text{e} \in \text{E} \tag{2}$$

$$T_e = \sum_{\forall \text{d} \in \text{D}} n_e^d \tag{3}$$

where, C_e^d is the amount of complementary data of the expression e in the database d, n_e^d is the number of images of the expression e in the database d, M is number of images of the expression with most samples, T_e is the number of images of the expression e considering all databases used during training, D.

2) Spatial Normalization: The spatial normalization comprises three steps: rotation correction, cropping, and resizing of the image to the expected input of the network. The rotation correction used in this work follows the same procedure explained in [30]. Essentially, it aligns all faces with the horizon based on the position of both eyes. After the rotation correction, all images are cropped in order to spatially normalize them. The cropping operation reduces the background part of the image to achieve the same aspect of the images expected by our models (given the employment of fine-tuning, as explained later). In the spatial normalization, each image is cropped according to the distance between the eyes, centralized in the midpoint between the eyes. The cropping area is determined by four boundaries: on the sides (left and right), the images are cropped based on 2.2 times the distance between the eyes for each side from the center; on the top, the boundary is determined by 2 times the distance between the eyes from the center; and on the bottom, it is 2.5 times the distance between the eyes from the center. These factors were empirically determined in order to have the input image similar to the images that were used to train the original VGG-Face. The resize was performed on the cropped images using a bilinear interpolation, resulting in down-sampled images of 256×256 pixels.

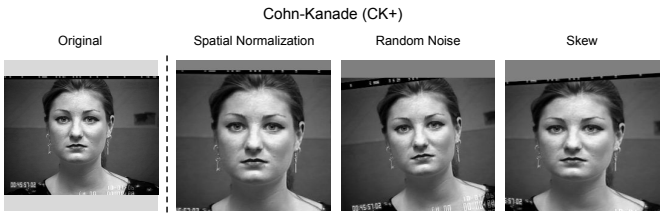

Fig. 2. Example of image from the CK+ database and its synthetic samples.

3) On-the-fly Data Augmentation: In addition to the offline data augmentation, some procedures are also applied on-the-fly during training phase: random crops and mirroring. These procedures are applied after the spatial normalization, where the input image is normalized to a 256×256 image. Then, this image is randomly cropped into a 227×227 image. The final image also can be randomly horizontally mirrored. In the test phase there is no data augmentation, but it is necessary to crop the image according to the trained input size. A centralized crop of 227×227 is performed, instead of random crops, and no mirroring is applied.

B. Convolutional Neural Network

The proposed system uses a Convolutional Neural Network to perform facial expression recognition: the VGG network, proposed by Simonyan and Zisserman [31]. They proposed the VGG CNN architecture that comes in two versions: VGG-16 and VGG-19 (i.e. 16 and 19 layers, respectively). In this work, the VGG-16 is used and referred to as VGG only. It has about 138 millions parameters and comprises 13 convolutional layers, followed by 3 fully-connected layers. The first two fully-connected layers have 4,096 outputs and the last has 2,622 outputs. Since this architecture was not originally proposed for the facial expression recognition problem, it is necessary to adapt the output layer to have 7 outputs (one for each expression) instead of the original 2,622 units.

For this network, two different initializations were evaluated: *i)* with random values (using Xavier algorithm [32]), and *ii)* with pre-trained weights. Pre-trained and randomly initialized networks usually differ in terms of the size of the initial learning rate. Random weights usually require higher base learning rate values to enable the gradient finding a good minimum. Pre-initialized weights usually require lower base learning rate values because they are already in the direction of the minimum. The latter training procedure is referred as fine-tuning.

For the fine-tuned models, the weights of the pre-trained VGG-Face model were loaded into the network. VGG-Face is the name of the model publicly released by the authors of the VGG network. This model was originally trained for the facial recognition task using a database of celebrity faces. As already explained, the last layer was changed from 2,622 to 7 units, therefore their weights had to be randomly initialized for all cases. In addition to this difference between the randomly initialized and fine-tuned models, the base learning rate is also

different. For the models with randomly initialized weights, all layers of the architecture were trained with the same initial base learning rate value. On the other hand, the learning rate of the output layer (i.e. the 7 outputs) is set to 10 times the base learning that is used in the previous pre-trained layers, in the case of fine-tuning. The random initialized network uses a base learning rate of 10^{-2} and the fine-tuned one uses as base learning rate the value that the original VGG-Face stopped, i.e. 10^{-4}. More details about the experimental setup is presented in the section IV.

IV. Experimental Methodology

In this work, an extensive experimentation with cross-database facial expression recognition is presented and the influence of fine-tuning a CNN pre-trained for a different task in a similar domain (face recognition) is investigated. For this investigation, seven widely used databases were chosen to train and test the models using the cross-database leave-one-out approach. The experimental methodology is detailed in the next subsections. Firstly, the databases used on the experimentation are presented. Subsequently, the experiments are described in details. After that, the metrics used on the experimentation are shown. Finally, the setup used during the experimentation is presented.

A. Databases

To achieve our goal, seven databases widely used in the literature were selected to perform the cross-database experiments: AR Face [15], CK+ [16], BU3DFE [17], JAFFE [18], MMI [19], RaFD [20], and KDEF [21] databases. Figure 3 shows the distribution of the expressions in each database. Moreover, Figure 2 shows examples of one subject of each database and the result of the offline data augmentation. Along with CK+, JAFFE, and AR Face databases, are available the files that contains the position of the eyes. The position of the eyes of MMI, RaFD, KDEF, and BU3DFE databases were manually annotated with aid of a face tracker algorithm.

1) CK+: The Extended Cohn–Kanade (CK+) database [16] consists of 100 university students aged from 18 to 30 years, resulting in 1,236 images. From those, 65% were female, 15% were African-American and 3% were Asian or Latino. The database comes from videos and each subject was instructed to perform expressions that begin and end with the neutral, i.e. expressions are posed.

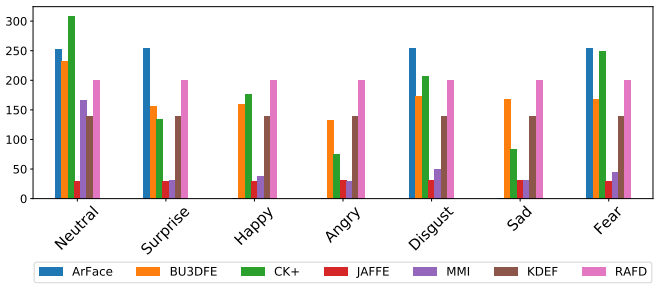

Fig. 3. Distribution of the expressions for all databases used.

2) JAFFE: The Japanese Female Facial Expression (JAFFE) database [18] contains 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.

3) MMI: The MMI database [19] contains video sessions with people showing emotions. In total, 32 subjects from the 235 sessions with labeled emotions were selected. From these sessions, frames with the frontal face of the subject showing the emotion (one of the six expressions or the neutral) were extracted, resulting in the 390 images used in the experiments.

4) RaFD: The Radboud Faces Database (RaFD) [20] consists of 67 models (including Caucasian males and females, Caucasian children, both boys and girls, and Moroccan Dutch males). In total, 1,407 images from this database in the six expressions plus the neutral were used in the experiments.

5) KDEF: The Karolinska Directed Emotional Faces (KDEF) [21] database consists of 70 actors (35 male, 35 female), aged from 20 to 30 years. Were used 980 pictures of human facial expressions from six emotions plus the neutral. Only the frontal face images were used in the experiments and the eyes position are annotated manually with face-tracker support.

6) BU3DFE: The Binghamton University 3D Facial Expression (BU-3DFE) database [17] contains 1,191 images from 58 female subjects, from several ethnicities, including White, Black, East-Asian, Indian and Hispanic Latino. This database was used only in the training phase because it is not complete (only with females).

7) AR Face: The AR Face database [15] contains frontal face images of 126 people (55.6% are male and 44.4% are female) over different facial expressions, lighting conditions and occlusion. No restrictions were applied to the participants in relation to clothing, hairstyles, makeup, etc. The AR Face does not have all expressions such as the other databases. Only three expressions (angry, happy, surprise) and the neutral were used, remaining 1,018 images in the database.

B. Experiment

Two methods were evaluated in this experiment: *i)* VGG with randomly initialized weights (hence VGG-Random), *ii)* fine-tuned VGG (hence VGG-FineTuning). All models were trained using Stochastic Gradient Descent (SGD) with a Step Down policy for the learning rate, decreasing it three times during the training (one for each of the three epochs). The maximum of three epochs was chosen because the models graphically showed convergence after these number of epochs in empirical experiments.

An extensive experimentation was designed to evaluate the different models trained in the proposed system. The experimentation comprises a cross-database leave-one-out approach. For this experiment, groups were created to extensively train and test each model. Each training set comprises a combination of six databases, always leaving one out. As BU3DFE and AR Face are always in the training sets, 5 groups were created in total. For each training set, the data

augmentation is performed and the expressions are balanced. The number of samples in each test set remains unchanged, i.e. the models are evaluated in the original database only with spatial normalization.

To reduce the influence of random factors in the experiments, each model combination was run 10 times with a different seed. The seeds were kept fixed among different methods within the same run. To reduce the randomness of the evaluation process, the algorithm used in the backpropagation of the weights were chosen to be deterministic. Note that even for the fine tuning there are some randomness in the process, as for example the weights of the output layer. The performance metrics of each method is presented in the results.

C. Metrics

Two performance metrics are reported: micro-averaged accuracy and macro-averaged accuracy. These two metrics were chosen because the micro-averaged accuracy is more commonly used in the literature, but it does not account for the unbalance of the classes in the databases. On the other hand, the macro-averaged accuracy takes the unbalance into account, which is the case as can be seen in the Figure 3.

In addition to the accuracy, a statistical analysis is performed to verify if the improvements were statistically significant. The paired *t*-test was used to estimate the significance of the pairwise comparisons considering the 10 runs. This test was performed for all methods used in the experiment. Differences were considered statistically significant for p-value < 0.01.

D. Setup

All the experiments were carried out using an Intel Core i7 4770 3.4 GHz with 16GB of RAM and a NVIDIA Tesla K40 with 12GB of memory. The environment of the experiments was Linux Ubuntu 14.04, with the NVIDIA CUDA Framework 7.5 and the cuDNN library 5.1.

The pre-processing (color conversion, data augmentation, and spatial normalization) was implemented using OpenCV and C++. The training and test phases were done using the NVIDIA fork of Caffe framework [33]. Some modifications were made in this version of the Caffe framework to ensure a deterministic behavior of the backproagation of the weights in order to be able to compare the results with the same seed. Basically, the convolutional layer of the Caffe framework was changed to ensure that deterministic algorithm were chosen during the backpropagation of the weights when using cuDNN (default behavior in non-deterministic).

Most of the works in the literature do not provide a way to reproduce their results, neither a detailed explanation of the conversion of the video databases. Therefore, in order to allow for reproducibility, we publicly released: *i)* a script to automatically convert video databases (e.g., CK+ and MMI) into the samples we used in this work, *ii)* a script to preprocess all databases in order to reproduce the same samples we used in the experimentation, *iii)* pre-trained models (the best of each method), and *iv)* a script to perform the inference and reproduce the results hereby reported.[1] Given these releases, we also expect to allow for fairer comparisons in the future.

V. Results and Discussions

This experiment evaluates two methods (VGG-FineTuning and VGG-Random) through an extensive experimentation with different groups of databases. Each group uses 6 databases for training and the remaining one for test. As two of these databases were used only during training, five of the most commonly used databases in the literature were used to test. Each group is named after the database used as test set. The results of this experiment are summarized in the Table I. As it can be seen, the VGG-FineTuning achieved, on average, the best performance, reporting up to 88.58% (CK+ database).

TABLE I
MICRO-AVERAGED ACCURACY AND STANDARD DEVIATION OF THE 10 RUNS FOR EACH OF THREE METHODS IN OUR FIVE DATABASE GROUPS.

Group	VGG Model (%)					
	FineTuning	Min	Max	Random	Min	Max
CK+	$\mathbf{88.58 \pm 0.43}$	87.8	89.1	78.09 ± 2.26	76.0	83.0
JAFFE	44.32 ± 2.45	40.4	50.2	$\mathbf{49.62 \pm 3.71}$	42.3	54.9
MMI	$\mathbf{67.03 \pm 1.66}$	64.1	69.5	55.64 ± 4.06	46.7	59.7
RaFD	$\mathbf{85.97 \pm 0.43}$	85.3	86.8	83.68 ± 2.74	78.4	87.1
KDEF	$\mathbf{72.55 \pm 0.72}$	71.3	73.7	69.03 ± 1.97	65.2	72.1

The results show VGG-FineTuning achieved better average accuracy in 4 out of 5 groups: CK+, MMI, KDEF, and RaFD. As it can be seen in the Table I, the results vary significantly between the groups, i.e. the performance of these models using the cross-database protocol is dependent on similarity of the test set conditions to the training set. This variation indicates that testing in a single database is not enough to assess the performance of a given model in the facial expression recognition task. There was only one database that the VGG-FineTuning did not perform better: JAFFE. It is important to note that JAFFE is a highly biased database in terms of gender and ethnicity, i.e. it comprises only Japanese female subjects. In addition, the results in the JAFFE database were the worst among the test groups. The result for the JAFFE database indicates that using fine-tuning may not always lead to the best results. Additionally to the variation between the databases, there is also another variation that is widely ignored in the literature caused by the random initialization. To measure this effect, 10 runs were performed. It can also be seen in the Table I that the fine-tuned models presented lower standard deviation when compared to the VGG-Random (e.g., 0.43 and 2.26, respectively, for the CK+), which indicates they tend to be more stable. Moreover, the JAFFE database also presented the highest standard deviation in the fine-tuned models with a variation of 9.85%. In addition, the MMI database got the highest variation for VGG-Random model: 13.07%. The statistical analysis performed between these two models show that only for the RaFD database the results did not show a significance, and all the others presented a p-value lower than

[1] https://github.com/viniz/facialexpressionrec

0.01, thus this shows that, in average, VGG-FineTuning is better. Finally, the results of the VGG-Random for the CK+ were computed using only 8 out of 10 runs, because two of them did not converge at all (with the same seeds used in the other methods for these runs). Therefore, in order to be fairer, this specific result (VGG-Random on the CK+ database) was calculated ignoring these 2 outliers. This shows that random initialization, as mentioned, tends to be more unstable.

Additionally, the VGG model was fine-tuned with freezing method (i.e. keeping the weights of the convolutional layers fixed and optimizing the last three fully connected layers), but the results about the same as the random initialization and therefore are not reported here.

Comparing results with the literature of cross-database facial expression recognition is very difficult. This is mostly because different authors can use different sets of databases and, for some cases (e.g. databases originally in video), it is hard to ensure the same images are being used across comparisons. Even though, some authors [23], [28] do perform such comparisons. Therefore, we present a comparison to show that in absolute terms, our method outperforms the results reported in the literature for most of the databases. The results presented in this section corresponds to micro-averaged accuracy, thats commonly used in the literature. Although, our results calculated with macro-averaged accuracy did not gets bigger difference between metrics, and giving less difference in VGG-FineTuning model. It is important to notice that RaFD and KDEF databases, as can be seen in Figure 3, are balanced, then did not show difference results between the two metrics. As it can be seen in the Table II, our models achieved state-of-the-art results for most of the databases used in our evaluation.

The Table II shows that our VGG-FineTuning models (pre-trained using the VGG-Face) present state-of-the-art results for 4 out of 5 databases: CK+, MMI, RaFD, and KDEF. For the CK+ test set, our models reported $88.58\% \pm 0.43$ of accuracy (best run with 89.1%), which represents a significant improvement (+23.77%) in the literature (64.81% [28]). For the MMI test set, our models reported $67.03\% \pm 1.66$ of accuracy, this result is 14.20% higher than [28] (using single frame results). For the RaFD test set, our models reported $85.97\% \pm 0.43$ of accuracy (best run with 86.8%). This result is 30.12% higher than the best result of the literature (55.85% [34]). Finally, although the KDEF have been extensively used in the intra-database protocol [35], there are no use of this database using the cross-database protocol. Therefore our results can be used as a baseline for future comparisons. Our models achieved $72.55\% \pm 0.72$ of accuracy (best run with 73.7%).

Looking deeper into the results of the MMI database, there are two facts worth mentioning. Firstly, the best results, [28] with CRF, were achieved using a method that considers the temporal relation between the frames (MMI is originally a video database). As our models process a single frame instead of a temporal sequence, it would be fairer to compare them with the other method the same author proposed that only process a single frame at a time. This model achieves 52.83% of accuracy (15.68% less than the other), which is 14.20% worse than ours. Secondly, in another perspective, considering our average and standard deviation, our model yields $67.03\% \pm 1.66$ for the MMI, which is on pair with the best results of [28]. Finally, in this context, we can state that our model also achieves state-of-the-art results for the MMI, especially considering a single frame at a time. For the JAFFE database, the best run of our model reported 50.23% of accuracy, which is higher (+1.56%) than the best result of the literature (48.67%). It is worth noticing that JAFFE also shows the worst results across the literature.

TABLE II
COMPARISON OF OUR VGG-FINETUNING MODEL WITH OTHER RESULTS IN THE LITERATURE IN TERMS OF MICRO-AVERAGED ACCURACY.

Group	Method	Train Database	Accuracy
CK+	da Silva and Pedrini [5]	JAFFE	48.20%
	da Silva and Pedrini [5]	BOSPHORUS	57.60%
	Zhang et al. [4]	MMI	61.20%
	Hasani et al. [28]	FERA/MMI	64.81%
	Hasani et al. [28] (CRF)	FERA/MMI	73.91%‡
	Mollahosseini et al. [7]	6 databases*	64.20%
	Our	**6 databases†**	$\mathbf{88.58\% \pm 0.43}$
JAFFE	Shan et al. [3]	CK	41.30%
	Ali et al. [34]	**RaFD**	**48.67%**
	da Silva and Pedrini [5]	CK	42.30%
	Our	6 databases†	$44.32\% \pm 2.45$
MMI	Shan et al. [3]	CK	51.10%
	Zhang et al. [4]	CK	66.90%
	Hasani et al. [28]	FERA/CK	52.83%
	Hasani et al. [28] (CRF)	FERA/CK	68.51%‡
	Mollahosseini et al. [7]	6 databases*	55.60%
	Our	**6 databases†**	$\mathbf{67.03\% \pm 1.66}$
RaFD	Ali et al. [34]	JAFFE	52.15%
	Ali et al. [34]	TFEID	55.85%
	Our	**6 databases†**	$\mathbf{85.97\% \pm 0.43}$
KDEF	**Our**	**6 databases†**	$\mathbf{72.55\% \pm 0.72}$

* Trained with all (MultiPIE, MMI, CK+, DISFA, FERA, SFEW, and FER2013) except the test set.
† Trained with all (CK+, JAFFE, MMI, RaFD, KDEF, BU3DFE, ARFace) except the test set.
‡ The result reported is not directly comparable with our results.

The results of the literature (Table II) also confirm that the performance of the models using the cross-database protocol varies according different databases, therefore it really seems inappropriate to generalize the performance of a model using a single database. All these comparisons should be carefully performed, since most of the methods of the literature do not release their models neither a way to reproduce their results.

VI. CONCLUSION

In this paper, we extensively investigated the use of a fine-tuned CNN architecture in the facial expression recognition problem using the cross-database protocol. In the proposed system, we fine-tuned the VGG network pre-trained in a face recognition database to recognize facial expressions considering seven different well-established databases in the literature. These datasets comprise more than 6,200 images

from subjects different ethnicities, genders, and ages in a variety of environments and in both spontaneous and posed facial expressions. Our results showed that employing fine-tuning to a CNN pre-trained on a similar domain is, on average, better than training from scratch. Moreover, fine-tuned models also yield more stable performance, i.e. with lower variance considering the intrinsic randomness of the initialization. In the comparison with the literature, the VGG-FineTuning models achieved state-of-the-art results for most of test sets in terms of absolute value, especially for the CK+ and RaFD database (88.58% and 85.97%, respectively) where there was a significant improvement in the literature (+23.77% and +30.12%, respectively). Although we cannot conclude the evaluated method is better than the methods of the literature due to the lack of standardization in the evaluation protocols, we can conclude the presented results are the state-of-the-art for cross-database facial expression recognition.

ACKNOWLEDGMENT

We would like to thank Fundo de Apoio a Pesquisa (FAP) of UFES for the support and CAPES for the scholarships. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

REFERENCES

[1] A. Mehrabian, "Communication without words," *Communication Theory*, pp. 193–200, 2008.

[2] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion." *Journal of personality and social psychology*, vol. 17, no. 2, p. 124, 1971.

[3] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009.

[4] X. Zhang, M. H. Mahoor, and S. M. Mavadati, "Facial expression recognition using l_p-norm mkl multiclass-svm," *Machine Vision and Applications*, vol. 26, no. 4, pp. 467–483, 2015.

[5] F. A. M. da Silva and H. Pedrini, "Effects of cultural characteristics on building an emotion classifier through facial expression analysis," *Journal of Electronic Imaging*, vol. 24, no. 2, pp. 023 015–023 015, 2015.

[6] A. T. Lopes, E. de Aguiar, A. F. D. Souza, and T. Oliveira-Santos, "Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order," *Pattern Recognition*, 2016.

[7] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Applications of Computer Vision (WACV), IEEE Winter Conference on*, 2016, pp. 1–10.

[8] P. Liu, S. Han, Z. Meng, and Y. Tong, "Facial expression recognition via a boosted deep belief network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1805–1812.

[9] P. Burkert, F. Trier, M. Z. Afzal, A. Dengel, and M. Liwicki, "Dexpression: Deep convolutional neural network for expression recognition," *arXiv preprint arXiv:1509.05371*, 2015.

[10] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 435–442.

[11] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 396–404.

[12] X. Zhao, X. Liang, L. Liu, T. Li, Y. Han, N. Vasconcelos, and S. Yan, "Peak-piloted deep network for facial expression recognition," in *ECCV*. Springer, 2016, pp. 425–442.

[13] G. Levi and T. Hassner, "Emotion recognition in the wild via convolutional neural networks and mapped binary patterns," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 503–510.

[14] H. Ding, S. K. Zhou, and R. Chellappa, "Facenet2expnet: Regularizing a deep face recognition net for expression recognition," *arXiv preprint arXiv:1609.06591*, 2016.

[15] A. M. Martinez, "The AR face database," *CVC technical report*, vol. 24, 1998.

[16] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and 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*, June 2010, pp. 94–101.

[17] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3d facial expression database for facial behavior research," in *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*. IEEE, 2006, pp. 211–216.

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

[19] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 5–pp.

[20] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. van Knippenberg, "Presentation and validation of the radboud faces database," *Cognition and emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.

[21] D. Lundqvist, A. Flykt, and A. Öhman, "The karolinska directed emotional faces (kdef)," *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, pp. 91–630, 1998.

[22] Y. Lv, Z. Feng, and C. Xu, "Facial expression recognition via deep learning," in *Smart Computing (SMARTCOMP), 2014 International Conference on*. IEEE, 2014, pp. 303–308.

[23] C. Mayer, M. Eggers, and B. Radig, "Cross-database evaluation for facial expression recognition," *Pattern recognition and image analysis*, vol. 24, no. 1, pp. 124–132, 2014.

[24] I. Song, H.-J. Kim, and P. B. Jeon, "Deep learning for real-time robust facial expression recognition on a smartphone," in *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, 2014, pp. 564–567.

[25] H. Y. Patil, A. G. Kothari, and K. M. Bhurchandi, "Expression invariant face recognition using local binary patterns and contourlet transform," *Optik-International Journal for Light and Electron Optics*, vol. 127, no. 5, pp. 2670–2678, 2016.

[26] Z. Wang, Q. Ruan, and G. An, "Facial expression recognition using sparse local fisher discriminant analysis," *Neurocomputing*, vol. 174, pp. 756–766, 2016.

[27] S. Arivazhagan, R. A. Priyadharshini, and S. Sowmiya, "Facial expression recognition based on local directional number pattern and anfis classifier," in *2014 International Conference on Communication and Network Technologies*, Dec 2014, pp. 62–67.

[28] B. Hasani and M. H. Mahoor, "Spatio-temporal facial expression recognition using convolutional neural networks and conditional random fields," *arXiv preprint arXiv:1703.06995*, 2017.

[29] P. Y. Simard, D. Steinkraus, J. C. Platt *et al.*, "Best practices for convolutional neural networks applied to visual document analysis." in *ICDAR*, vol. 3. Citeseer, 2003, pp. 958–962.

[30] A. T. Lopes, E. de Aguiar, and T. Oliveira-Santos, "A facial expression recognition system using convolutional networks," in *Graphics, Patterns and Images (SIBGRAPI), 28th Conference on*, 2015, pp. 273–280.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[33] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[34] G. Ali, M. A. Iqbal, and T.-S. Choi, "Boosted nne collections for multicultural facial expression recognition," *Pattern Recognition*, vol. 55, pp. 14–27, 2016.

[35] M. Xu, W. Cheng, Q. Zhao, L. Ma, and F. Xu, "Facial expression recognition based on transfer learning from deep convolutional networks," in *Natural Computation (ICNC), 2015 11th International Conference on*. IEEE, 2015, pp. 702–708.