Expert Systems With Applications 136 (2019) 1–11

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Comparing ensemble strategies for deep learning: An application to facial expression recognition

Alessandro Renda[a,b], Marco Barsacchi[a,b], Alessio Bechini[a], Francesco Marcelloni[a,*]

[a] *University of Pisa, Dept. of Information Engineering, Largo L. Lazzarino, Pisa, 56122, Italy*
[b] *University of Florence, Dept. of Information Engineering, Via di Santa Marta, 3, Firenze, 50139, Italy*

ARTICLE INFO

Article history:
Received 6 September 2018
Revised 18 May 2019
Accepted 13 June 2019
Available online 14 June 2019

Keywords:
Facial Expression Recognition
Convolutional Neural Networks
Ensemble learning
Ensemble construction

ABSTRACT

Recent works have shown that Convolutional Neural Networks (CNNs), because of their effectiveness in feature extraction and classification tasks, are suitable tools to address the Facial Expression Recognition (FER) problem. Further, it has been pointed out how ensembles of CNNs allow improving classification accuracy. Nevertheless, a detailed experimental analysis on how ensembles of CNNs could be effectively generated in the FER context has not been performed yet, although it would have considerable value for improving the results obtained in the FER task. This paper aims to present an extensive investigation on different aspects of the ensemble generation, focusing on the factors that influence the classification accuracy on the FER context. In particular, we evaluate several strategies for the ensemble generation, different aggregation schemes, and the dependence upon the number of base classifiers in the ensemble. The final objective is to provide some indications for building up effective ensembles of CNNs. Specifically, we observed that exploiting different sources of variability is crucial for the improvement of the overall accuracy. To this aim, pre-processing and pre-training procedures are able to provide a satisfactory variability across the base classifiers, while the use of different seeds does not appear as an effective solution. Bagging ensures a high ensemble gain, but the overall accuracy is limited by poor-performing base classifiers. The impact of increasing the ensemble size specifically depends on the adopted strategy, but also in the best case the performance gain obtained by involving additional base classifiers becomes not significant beyond a certain limit size, thus suggesting to avoid very large ensembles. Finally, the classic averaging voting proves to be an appropriate aggregation scheme, achieving accuracy values comparable to or slightly better than the other experimented operators.

1. Introduction

As facial expressions play vital roles in human interactions and nonverbal communications, Facial Expression Recognition (FER) is of crucial importance to the development of interactive computer systems. A facial expression represents a *signal* that humans use, intentionally or otherwise, in order to convey a *message*, i.e. an emotion, an affective state, or a health condition. In 1971, Ekman and Friesen (1971) demonstrated that facial expressions of emotion are universal. The study was carried out on both literate and preliterate cultures: the universality of the human way of expressing an emotion is supposed to be an evolutionary, biological fact, not depending on the specific culture. This finding allows modern Computer Vision studies to focus on the signal (facial expression) in order to analyze the message (emotion), giving rise to plentiful applications in different fields, ranging from Human Computer Interaction to Data Analytics (Martinez & Valstar, 2016). Furthermore, FER has been automated and several machine learning algorithms have been specifically proposed for this task (Gross & Brajovic, 2003; Martinez & Valstar, 2016; Pramerdorfer & Kampel, 2016; Zhang, Mahoor, & Mavadati, 2015a). Effective solutions for expert systems targeted at solving the FER problem have to be extensively investigated. Automated FER approaches attempt to classify faces in a given single image as one of the six basic emotions, namely anger, disgust, fear, happiness, sadness, and surprise.

A related and relatively recent topic in the area of research on *affective computing* consists in emotion recognition from videos. On the one hand, video data pose additional challenges to the task of emotion recognition compared to static images, e.g. the quick and variable dynamics between the beginning of the expression (*onset*), its peak, and its vanishing (*offset*). On the other hand, the amount of information provided by the sequence of correlated frames and, in some instances, by the associated speech, enables a variety of

[*] Corresponding author.
E-mail addresses: alessandro.renda@unifi.it (A. Renda), marco.barsacchi@unifi.it (M. Barsacchi), alessio.bechini@unipi.it (A. Bechini), francesco.marcelloni@unipi.it (F. Marcelloni).

https://doi.org/10.1016/j.eswa.2019.06.025
0957-4174/© 2019 Elsevier Ltd. All rights reserved.

automated methods for robust features extraction and classification in a multi-modalities setting. In this work we focus on FER from static images, rather than from video sequences: the nature of video data and underlying assumptions (multimodalities of audiovisual data, correlation among sequential frames) make it a different, although closely related, and more challenging topic than static image analysis. Nevertheless, many recent works attest that FER on static images is still an active research area and advances in this field may have a positive impact in the field of emotion recognition from videos too.

In this context, researchers have collected several annotated face databases both in spontaneous uncontrolled setting (Dhall, Goecke, Joshi, Sikka, & Gedeon, 2014) and in more strictly controlled environments (Gross, Matthews, Cohn, Kanade, & Baker, 2010; Lucey et al., 2010; Lyons, Akamatsu, Kamachi, & Gyoba, 1998). Images acquired in controlled conditions (or lab-conditions) consist in posed expressions of frontal faces, with standard illumination and background conditions. Nowadays, emotion recognition in this scenario is considered a solved problem and is primarily used for the proof of concept of features extraction and classification methods (Pramerdorfer & Kampel, 2016; Sariyanidi, Gunes, & Cavallaro, 2015). Indeed, several works have shown that a recognition rate above 90% can be achieved under these conditions (Dornaika, Moujahid, & Raducanu, 2013; Mahersia & Hamrouni, 2015) and, in more recent proposals, accuracy values close to 100% on well-known benchmark datasets have been reached (Liang, Liang, Yu, & Zhang, 2019; Xie, Hu, & Wu, 2019). Within the wide assortment of classical machine learning algorithms, several of them - notably Support Vector Machines and Bayesian classifiers - have proved to be able to classify posed facial expressions generated in a controlled environment. Nevertheless, these approaches fell short of the generalization capability. FER under naturalistic conditions, often referred to as *in-the-wild*, is the scenario of interest for what concerns the above mentioned applications (Dhall et al., 2014). The factors of variation that make this a harder problem are the following: subtlety of spontaneous expressions, head pose, illumination, and occlusions.

A standard algorithmic pipeline to address the FER problem on static images relies on the crucial step of feature extraction. Traditional approaches consist in determining features by hand through mathematical descriptors (e.g. Gabor filter, Local Binary Pattern, Scale Invariant Feature Transform) (Hussain, Khan, Nazir, & Iqbal, 2012), or using facial landmarking (Tie & Guan, 2013). Exploiting hand-crafted features has proved to be inadequate for the in-the-wild task: an optimal feature extractor should provide information useful for the classification step, being robust to the above mentioned nuisance factors. Recent research tried to investigate on the possibility to learn features directly from data (Hertel, Barth, Kster, & Martinetz, 2015). In computer vision, the most popular models used for this purpose are the CNNs, and new practical methodologies have been studied for their employment in modern expert systems (Han, Liu, & Fan, 2018).

The origin of CNNs dates back to the '80s; nevertheless, they have been largely ignored from the mainstream computer vision and machine learning communities, up to the ImageNet competition in 2012. Their resurgence, as well as the impressive results they achieve, can be ascribed to the efficient use of GPUs, ReLUs, dropout, and new data augmentation techniques (LeCun, Bengio, & Hinton, 2015). CNNs have thus revolutionized the field of computer vision, and nowadays they are the dominant approach for all recognition and detection tasks, even approaching the human performance on specific cases.

The theoretical advantages of using deep architectures have been highlighted in the literature (Bengio, 2009). On the one hand, cognitive processes in humans seem to have a deep structure, with different levels of representation and abstraction: CNNs are inspired to the mammalian vision system and, in particular, to the bidimensional structure of visual cortex and to the relative biological neurons. On the other hand, too shallow architectures fail in representing the desired function with a reasonable number of parameters: the required number of units might grow exponentially if the depth is reduced. Furthermore, it has been shown that the spatial regions of the input facial image, which maximally excite neurons in the hidden layers of the proposed convolutional networks, correspond to the Facial Action Units described by Ekman (Khorrami, Paine, & Huang, 2015), that is, the network is able to learn relevant high-level features.

It is worth underlining that research in the FER task is hindered by the lack of a large amount of labeled training data, typically necessary in current deep learning approaches. Indeed, unlike visual object databases such as imageNet, existing FER databases often have a limited number of subjects, few sample images or videos per expression, or small variations between sets, hampering the neural network training procedure. For instance FER2013, which is one of the largest databases built so far, consists of 35,887 images of different subjects and, yet, only 547 of them portray disgust. Gathering and annotating new data is often a difficult, expensive, and time consuming task. The challenge is indeed to find alternative methods in order to improve the performance of automatic FER systems.

The reported average human accuracy on FER-2013 dataset is 65%. With the work presented in (Tang, 2013), Tang won the machine learning competition in the ICML 2013 Challenges in Representation Learning, achieving a test accuracy of 71.2% using a CNN with L2-SVM loss. This figure has been further improved in the recent years: Hereafter, we recall some of the most significant works along with their performance, i.e. the accuracy obtained over the test set. In (Kim, Dong, Roh, Kim, & Lee, 2016a), Kim et al. achieved a 73.73% test accuracy by means of an ensemble of CNNs that uses both aligned and non-aligned images: the key factor for the performance improvement is represented by a pre-processing (alignment) operation carried out by yet another Deep Convolutional Network that learns the proper mapping. In (Connie, Al-Shabi, Cheah, & Goh, 2017), Connie et al. proposed a model that combines SIFT features and CNN features: the aggregation of three models let them achieve a 73.4% accuracy. Excellent results have been obtained also by exploiting diversified learning information: Zhang, Luo, Loy, and Tang (2015b) reached a 75.1% test accuracy by fusing training data from multiple sources.

The work presented in this paper aims to propose general, more efficient procedures to construct ensembles of CNNs for the FER task: looking for general results, the relative experimentation must focus on the *basic* FER ability of each single network, independently of any possible ancillary geometric pre-processing action. However, it has been shown that such a pre-processing is crucial in getting top accuracies (Kim et al., 2016a). As a consequence, the performance obtained through a basic ensembling procedure is not directly comparable with the most precise state-of-the-art systems, because of the different nature of their building blocks. Nevertheless, even if our studied configurations adopt neither face-alignment techniques (or tricks of this kind) nor specific particular features, their delivered accuracy closely trails the top performing works.

Ensemble solutions are widely exploited in neural networks, with the aim of boosting classification performance. Giacinto et al. highlight the keypoint on network ensembles for image classification purposes (Giacinto & Roli, 2001): Beyond general theoretical analysis (Brown, Wyatt, Harris, & Yao, 2005), experimental evidence exists showing that an ensemble can outperform the best single neural network, provided that the networks make *different* errors. From this perspective, the task of producing error-independent networks is not trivial, mainly because of the weight space symmetry. Two approaches can be exploited to design ensembles of neural networks: *implicit* and *explicit* methods. The former consists in directly creating error-independent neural networks by forcing diversity among them; the latter consists in producing a large set of base classifiers and selecting the optimal subset with respect to a given measure of the error diversity. In our work we focus on *implicit* ensemble design strategy. The most commonly adopted strategies to create heterogeneous ensembles are: i) varying the initial random weights, ii) varying the network architecture, iii) varying the network type, and iv) varying the training data (Giacinto & Roli, 2001). The way the networks outputs are aggregated to produce a final output represents another central issue in the ensembling approach: the most straightforward strategies for combining outputs computed by base classifiers are Average and Majority voting (Ponti Jr, 2011).

The network size represents a key factor for the learning dynamics: in fact, deeper models are less sensitive to randomness in the initialization and training procedure, leading to a less sparse distribution of loss in multiple repetitions (Choromanska, Henaff, Mathieu, Arous, & LeCun, 2015). As a consequence, ensemble learning has proven to be more effective when using shallow networks (Ju, Bibaut, & van der Laan, 2018) since the network high sensitivity to initialization and training most likely results in different local minima.

Recent proposals in the field of expert and intelligent systems have mainly focused on devising more sophisticated ensemble design and fusion strategies for the FER problem. Liu and Zhang proposed a two-step ensemble framework in the context of granular computing (Liu & Zhang, 2019). In their approach, ensembles can be viewed as information granules; a further level of ensemble and information fusion completes the granular architecture with a coarser level of granularity. The approach, however, is validated using a training dataset of only 344 instances, on which deep learning architectures fail in obtaining competitive performance. In (Gan, Chen, & Xu, 2019), authors combined eight neural networks obtained by training a CNN architecture on a training set with different perturbations of soft-labels, which are supposed to capture latent similarity and mixture among different facial expressions. Hierarchical committees of CNNs have been introduced as well (Kim, Roh, Dong, & Lee, 2016b), and in this case the single decisions are fused according to a multi-level structure. Finally, Wen et al. proposed a probability-based fusion rule to combine diverse base classifiers obtained by varying initialization of parameters and hyperparameters of a CNN architecture (Wen et al., 2017). None of these works, however, has improved state-of-the-art performance on the FER-2013 dataset and disentangled the factors that influence performance in ensemble of neural networks.

In this paper, in order to find efficient procedures for the ensemble construction in FER problems we evaluate the effectiveness of several strategies in exploiting the sensitivity of shallow networks used as basic classifiers. We fix the classifier type, namely CNN, and the network architecture using a model that can be considered shallow if compared to modern very deep architectures such as VGG (Simonyan & Zisserman, 2014), Inception (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016), and ResNet (He, Zhang, Ren, & Sun, 2016). As the ensemble performance is tightly related to the diversity of the classifiers making up the ensemble, particular care must be placed in the selection of the strategy used to generate the base classifiers. We present an analysis of various techniques for generating diversity within ensembles of base networks, varying training data and weight initialization. Furthermore, we compare different fusion schemes for merging the outputs of the base classifiers. We also investigate whether it is appropriate to use aggregation methods other than the common Average and Majority Voting. To this aim, we experiment with different Ordered Weighted Averaging (OWA) operators (Fodor, Marichal, & Roubens, 1995; Yager, 1993). The final objective is to provide some indications for building up effective ensembles of CNNs.

Although ensemble learning has been widely employed in FER and in other similar contexts, an experimental analysis of the aspects that influence its performance with CNNs as base classifiers has never been carried out. The main contributions of this study can be summarized as follows:

- Achieving competitive performance for the in-the-wild emotion recognition, avoiding to use hand-crafted features and adopting simple design strategies for the ensemble of CNNs;
- Shedding light on the factors that influence the performance, in terms of recognition accuracy, of such ensembles. Notably, the extensive analysis aims to compare different simple strategies for generating diversity among base classifiers, and different aggregation schemes for combining the prediction at the decision fusion level, and further to investigate the dependence of the performance upon the ensemble size.

The paper is organized as follows: In Section 2 we provide the background about CNNs. In Section 3 we describe the approaches for designing different ensembles of CNNs and the adopted fusion strategies. Section 4 presents the experimental setup: we describe the datasets used in the present work and the details of our implementation. In Section 5 we show and discuss the experimental results. Section 6 draws conclusion remarks.

2. Convolutional neural networks background

CNN (LeCun, Kavukcuoglu, & Farabet, 2010) is a type of feed-forward artificial neural network. The building blocks of a convolutional layer are the following: (i) The *linear convolutional stage* performs the convolution operation between a kernel, or filter, and an input bi-dimensional array; (ii) the *non-linear activation stage* consists in the pointwise application of a non-linear function; (iii) the *pooling stage* computes a summary statistic of a group of input neurons. Thanks to its architecture, CNN can take into account the spatial structure of the input: this is a desired property when the input layer has a known topology, e.g. the 2D grid of pixels that constitutes an image. Moreover, CNNs require fewer parameters than fully-connected networks. As a consequence, CNNs are faster to train and less prone to overfitting. Further, deeper models can be designed.

When a network is trained from scratch, random values are initially assigned to the parameters (weights w). Then, the gradient-based learning procedure iteratively updates the weights w by using an optimization algorithm, a form of stochastic gradient descent (Goodfellow, Bengio, & Courville, 2016). In its basic implementation, given the learning rate ε and a cost function f that we aim to minimize by modifying the parameters w, the weight update takes place according to the formula $w \leftarrow w - \Delta w$, where $\Delta w = \varepsilon \nabla_w f(w)$. Each iteration is performed on a fixed number of images (batch size) and requires two steps: a forward step and a backward step. During the forward step, the network maps each input sample to an output and produces a scalar cost, i.e. the discrepancy between the predicted output and the expected output. The backward step computes the gradient of the scalar cost with respect to the weights of the network. This procedure is efficiently performed by means of the backpropagation algorithm (LeCun, Bottou, Orr, & Müller, 1998). The training process is repeated for a given number of epochs: at each epoch the input pipeline covers the entire training set.

The validation set is used during the training to periodically evaluate the performance of the model on previously unseen examples. The model hyperparameters are tuned to maximize the validation accuracy (or minimize the validation loss) and to avoid overfitting. In this scenario, several techniques are typically adopted to reduce the discrepancy between training and validation errors:

- *Dropout* (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014): It reduces the co-adaption between units by randomly turning off with probability p each unit of a layer at every training step;
- *Data augmentation*: It consists in artificially increasing the size of the training set by applying one or more transformations in the input image domain, provided that the *transformed image* and *original label* mapping is preserved. Such an approach represents a convenient and efficient trick whenever gathering and annotating more training data is expensive, unfeasible, or time-consuming. Although violating the i.i.d. assumptions over the data generating process (i.e. examples should be *independent* and *identically distributed*), which let us to estimate the *test error* from the *training error* (Goodfellow et al., 2016), data augmentation is widely adopted in practical cases. Indeed, it has proven to be effective in improving generalization capability of deep learning models in several application areas, ranging from image classification (Han et al., 2018) to speech recognition (Jaitly & Hinton, 2013);
- *Weight regularization*: It typically consists in penalizing the magnitude of the weight by adding a term to the loss function. If L2 regularization is chosen, the loss function $J(\boldsymbol{w})$ becomes $J(\boldsymbol{w})^* = J(\boldsymbol{w}) + \lambda \boldsymbol{w}^T \boldsymbol{w}$, with λ as the weight decay factor.

3. Ensemble design

In this section, we first describe the different ensemble design strategies to investigate, pointing out how variability among the base networks is generated, and then we show the aggregation methods adopted in our experiments.

3.1. Ensemble design strategies

In our investigation, we take into account four different ensemble design strategies, selected to consider the main possible ways to address diversity in the construction of the base classifiers:

1. The first strategy, denoted as *Seed Strategy (SE)*, exploits a plain random initialization of the weights;
2. The second strategy, proposed in (Kim et al., 2016a) and denoted as *Preprocessing Strategy (PS)*, produces an ensemble by combining networks generated by different seeds and different image preprocessing functions;
3. The third strategy, denoted as *Pretraining Strategy (PT)*, exploits other, albeit small, datasets: it uses a combination of different preprocessing functions and weight initialization obtained by pretraining the network on datasets different from the one used in the training phase;
4. The fourth strategy, denoted as *Bagging Strategy (BS)*, uses bagging to generate variability among the base networks.

For each strategy, we use an ensemble of nine networks. In the following, we describe in detail the four strategies.

3.1.1. The seed strategy

The first strategy produces networks with the same configuration by simply varying the seed that initializes the pseudorandom number generator. Since the pseudorandom number generator determines how the network is generated by affecting the shuffle of the dataset, dropout, data augmentation, and initial distribution of the weights, different seeds produce different networks. SE is the simplest and, to the best of our knowledge, the most commonly used strategy in the case of the Deep Convolutional Neural Networks. The resulting ensemble structure consists of nine base classifiers obtained by setting nine different integers as the seeds for the learning procedure.

3.1.2. The preprocessing strategy

PS is defined according to the work by Kim et al. (2016a): It combines three different seeds with three preprocessing strategies that introduce variability on training data. In particular, the preprocessing strategies consist of, respectively, keeping the images unchanged (default), modifying the images by histogram equalization (histEq), and preprocessing them by using illumination normalization (iNor). The illumination normalization technique tries to compensate for illumination-induced variations in images (Gross & Brajovic, 2003) by smoothing with an isotropic diffusion technique. Histogram equalization is a fairly standard approach aimed at achieving contrast enhancement (Gonzalez & Woods, 2006). The resulting ensemble structure consists of nine base classifiers obtained by setting three different integers as the seeds for each of the three preprocessing techniques.

3.1.3. The pretraining strategy

PT is a modification of PS obtained by pursuing a higher internetwork variability: the combination of three seeds and three preprocessing methods is kept, but weights are initialized in three different ways, coherently with the seeds, by pretraining networks on other facial expression datasets. In particular, the three different initial conditions are: i) random initial distribution of the weights, ii) parameter configuration obtained by pretraining the network on the Extended Cohn–Kanade (CK+) dataset (Lucey et al., 2010), and iii) parameter configuration obtained by pretraining the network on the Static Facial Expression in the Wild (SFEW) dataset (Dhall et al., 2014; Dhall, Goecke, Lucey, & Gedeon, 2012). When training on FER2013, i.e. the target dataset described in Section 4.1, the initial weights of the convolutional layers are the same as obtained in pretraining, whereas the top fully connected layers are trained from scratch. This idea relies on the assumption that each of the datasets, described in Section 4.1, has a unique fingerprint, for example, typical illumination and pose condition. Since CK+ consists of posed and lab-controlled images, while SFEW is by definition in-the-wild, we expect that these two datasets can fulfill the purpose. The resulting ensemble structure consists of nine base classifiers obtained by initializing the weights according to three pretraining options (and setting three different integers as the seeds) for each of the three preprocessing techniques.

3.1.4. The bagging strategy

The fourth strategy exploits bagging to generate a different dataset for each network in the ensemble (Breiman, 1996), still keeping all the remaining random components fixed, using a single seed. Despite the popularity of this approach in the machine learning community for the generation of ensembles of classifiers, it is not widely applied in Deep Learning. The resulting ensemble structure consists of nine base classifiers obtained by setting nine different integers as the seeds for the random operation of sampling with replacement.

3.2. Aggregation strategies

The output layer of our CNN is the well-known “softmax” layer. Let K be the number of the neurons in this layer. The output value $\sigma_j(\mathbf{z})$ of neuron j is computed as:

$$\sigma_j(\mathbf{z}) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

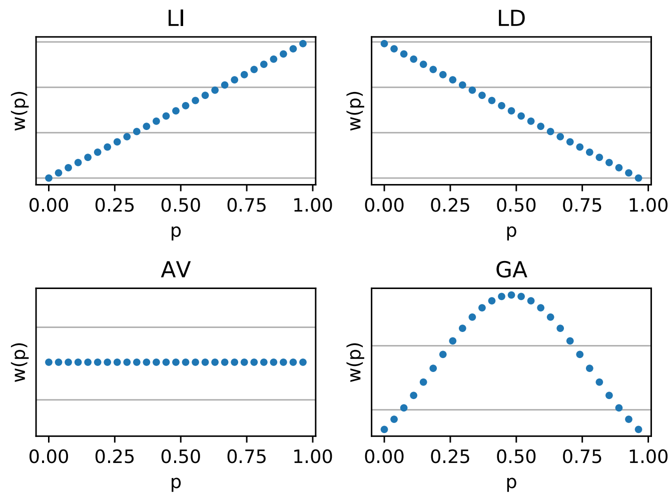

Fig. 1. The stress functions used to generate the weight vectors for OWA operators.

where z is a K-dimensional vector. The output values $\sigma_j(\mathbf{z})$ satisfy the following properties:

- $\sigma_j(\mathbf{z}) : \mathbb{R}^K \to [0, 1]$
- $\sum_{j=1}^{K} \sigma_j(\mathbf{z}) = 1$

Softmax is the default choice for neural networks in multi-class classification problems since it can be interpreted as the probability distribution over the classes.

Given the FER2013 test-set images, the evaluation step produces a matrix $P \times K \times D$, where P is the number of base classifiers, K is the number of classes (emotions), D is the number of images. Ensemble learning aims to combine the outputs of multiple classifiers in order to have a final prediction matrix of size $1 \times K \times D$. The class with the highest σ value represents the predicted label.

Our analysis focuses further on how to combine these pieces of information from the base classifiers in the ensemble. Average and Majority voting are the aggregation schemes typically adopted in ensembles of neural networks: majority voting selects the class that obtain the highest number of votes, and average voting can be interpreted as a special case of OWA operators.

Given a vector of P non-negative weights z_i such that $\sum_i z_i = 1$, the OWA aggregator $F^{(P)}$ applied to a vector $\mathbf{x} \in \mathbb{R}^P$ corresponds to:

$$F(\mathbf{x}) = \sum_{i=1}^{P} z_i x_{(i)}$$

with $x_{(1)} \leq \ldots \leq x_{(i)} \leq \ldots \leq x_{(P)}$. The final prediction for a given image can be interpreted as the combination, parametrized by the vector $\mathbf{z}$, of the decision profile matrix $P \times K$. The i-th row of this matrix represents the softmax output of the i-th base classifier, and the j-th column represents the posterior probabilities for class j from different classifiers.

As suggested in (Yager, 2007), the weight vector $\mathbf{z}$ could be obtained through the definition of a stress function: it let users determine what argument values should be weighted more than the others in the aggregation. Fig. 1 shows the stress functions adopted in this work. In our work, we follow the procedure proposed in (Alajlan, Bazi, AlHichri, Melgani, & Yager, 2013) in order to obtain the final predictions by using OWA operators. Indeed, we adopt the following aggregation techniques:

- Majority Voting
- OWA operators with the following stress functions:
  - AV: Average
  - LD: Linear Decreasing
  - LI: Linear Increasing
  - GA: Gaussian

Table 1
Number of images per class in the three datasets used in the experiments.

Label	FER-2013	CK+	SFEW
Neutral	6197	327	228
Anger	4945	135	255
Disgust	547	177	75
Fear	5121	75	124
Happiness	8988	207	256
Sadness	6076	84	234
Surprise	4001	249	149
Contempt	–	54	–

Fig. 2. Two examples of FER-2013 images.

4. Experimental set-up

In this section we describe the datasets used in the experiments and the details of our implementation.

4.1. Facial expression datasets

The experiments were carried out on three widely used facial expression datasets: the Facial Expression Recognition 2013 (FER-2013) dataset (Goodfellow et al., 2015), the Extended Cohn–Kanade (CK+) dataset (Lucey et al., 2010), and the Static Facial Expression in the Wild (SFEW) dataset (Dhall et al., 2014; Dhall et al., 2012). CK+ and SFEW have been used in the pretraining procedure only, while FER-2013 have been used for training and test. It represents a proper choice for our investigation because it consists in the largest available collection of *uncorrelated* in-the-wild images. Other datasets (e.g. MMI Valstar & Pantic, 2010, BP4D/BP4D+ Zhang et al., 2014; Zhang et al., 2016, JAFFE Lyons et al., 1998, CMU Multi-PIE Gross et al., 2010) have not been considered for the evaluation of our approach because of one or more of the following reasons: i) Consisting in collections of video clips, thus resulting in highly correlated examples, ii) having limited number of subjects, iii) consisting in posed expressions or controlled pictures.

Table 1 shows the number of images per each of the six basic expressions and neutral faces in each dataset used in our experiments.

4.1.1. The FER-2013 dataset

The classification accuracy on the FER2013 dataset represents the performance measure of the models used in the present work. Since the FER2013 dataset is one of the largest collections of *in the wild* facial images, it can be used as a proxy for FER task in real world condition. It has been released in 2013 for a machine learning contest held as part of the ICML workshop “Challenges in Representation Learning”. The official split of the FER-2013 dataset was used in the present work. The only modification was the removal of 11 “black” images (zero in all the pixels). This resulted in 28,699 images as training set, 3588 images as validation set, 3589 as test set. The provided face crops have been used with no additional detection stage. Two example images are shown in Fig. 2.

Fig. 3. Two examples of CK+ images after face detection with openCV.

4.1.2. The pretraining datasets

The application of the pretraining strategy (PS) asks for specific, different datasets. Hereafter we provide an accurate description of the two used in our experiments.

The *CK+ dataset* consists of 593 sequences from 123 subjects. Only 327 sequences of them are annotated with the labels reported in Table 1. In the present work, four images were selected from each sequence, according to the protocol used in (Khorrami et al., 2015): This resulted in 1308 images. As shown in Fig. 3, images have been obtained in lab-condition. The 1308 images of the dataset were split in training and validation sets. The unique constraint was that images of the same subject belonged to the same set. In order to accomplish this, an 1/10 fraction of the subjects was sampled to generate the validation set. This resulted in 1208 images for the training set and 100 images for the validation set. Faces were detected using Haar Cascade frontal face detector from OpenCV. Images were then re-sized to 162 × 162 and converted to gray-scale.

The *SFEW dataset* consists of images with temporal facial expressions, acted facial expressions in the wild, extracted from movies. The images are aligned as shown in Fig. 4. Even when using aligned datasets, FER is a difficult task because of variation in head pose, age, occlusions, focus and real world illumination condition. The original split of the dataset consists of 890 images for training set and 431 images for validation set. For the purposes of the present work a new split was adopted in order to increase the training set size. A tenth of the entire dataset was sampled to generate the validation set. This resulted in 1036 training images and 285 validation images. For consistence with the CK dataset, the images were then converted to gray-scale and resized from 143 × 181 to 162 × 162. The resize operation introduces an aspect ratio distortion, as can be noticed by comparing Fig. 4(b) and (c).

4.2. Adopted model and parametrization

Input data have been zero-centered after the application of the input transformation described in Section 4.1: A global mean value μ, and a global standard deviation value σ have been evaluated over each of the training sets of interest. The normalization step has been performed by subtracting μ and dividing by σ. The transformation has been subsequently applied to every training, validation, and test image.

4.2.1. The CNN architecture

For all experiments presented in this paper, we used a classical feed-forward CNN, inspired by the work in (Kim et al., 2016a). The architecture is illustrated in Fig. 5: The input layer is followed by three convolutional and max pooling layers with respectively 32, 32, and 64 feature maps. A fully connected layer of 1024 neurons precedes the output layer composed by 7 units, i.e. the seven classes of emotion considered in the FER-2013 database. The total network depth is 5 and the total number of trainable parameters is 2,436,007. The rectifier (ReLU) is used as non-linear activation function, for both convolutional and fully connected layers. Softmax activation is used in the output layer. Batch normalization (Ioffe & Szegedy, 2015) is applied after every convolutional and fully connected layer; zero-padding is used in the convolutional layers in order to preserve the original size of the input. The max-pooling layer consists of an overlapping kernel of size 3 × 3 and stride 2 × 2 that resulted in a size halving. Dropout layer is added after the fully connected hidden layer, with drop-probability of 0.15.

4.2.2. The training and inference on the FER2013 dataset

The learning algorithm adopted in the experiments is the stochastic gradient descent with momentum of 0.9. The loss function is composed by a cross-entropy term and an L2 regularization term with a weight decay factor of 0.0001. The learning rate schedule is determined by the array of step boundaries [12000, 18000, 24000, 30000, 36000] and the array of learning rate values [0.1, 0.05, 0.025, 0.0125, 0.00625, 0.003125]. The batch size is 200. After 300 epochs, the learning process stops if the validation accuracy does not increase for consecutive 20 epochs.

Data augmentation is performed during the training: A random crop of size 48 × 48 is selected after zero-padding the original images from 48 × 48 to 54 × 54. Every image is then flipped horizontally with probability 0.5. During the evaluation over the validation and test sets, a ten-crop oversampling is performed as suggested in (Krizhevsky, Sutskever, & Hinton, 2012): after zero-padding at 54 × 54, 5 crops are selected: the central one and four patches of size 48 × 48 at the corners. The five images are then horizontally flipped, resulting in 10 versions of the same image. The softmax outputs of the ten images are then averaged to obtain the final prediction.

4.2.3. The pre-training on the CK+ and SFEW datasets

Due to the limited size of the CK+ and SFEW datasets, an additional data augmentation step is performed. Images are resized from 162 × 162 to 54 × 54. Each image is rotated by a random angle in the range $[-15^\circ, +15^\circ]$ and horizontally flipped with probability 1/2. Then, a random crop of variable size is selected in order to achieve a random scaling of the input image. The scaling factor is drawn from a uniform distribution in the range [0.852, 1] and generate a patch whose size is in the range from 46 × 46 to 54 × 54. The image is then re-sized to 48 × 48 to make it consistent with the chosen architecture for the network. The weight decay factor

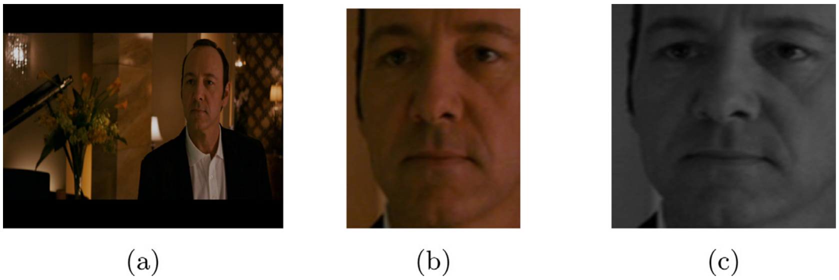

Fig. 4. Example of SFEW Image: a) Image of the original dataset; b) Image of dataset Aligned-SFEW; c) Image after grayscale transformation and resizing.

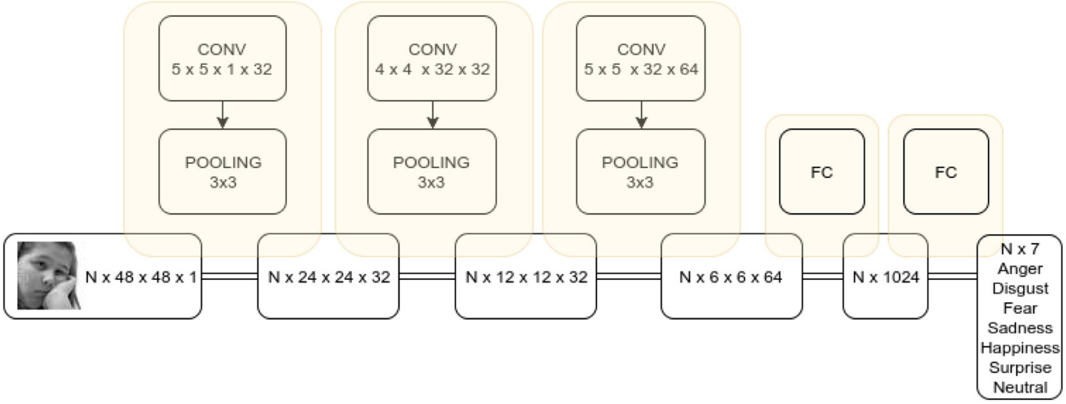

Fig. 5. Scheme of the adopted network architecture.

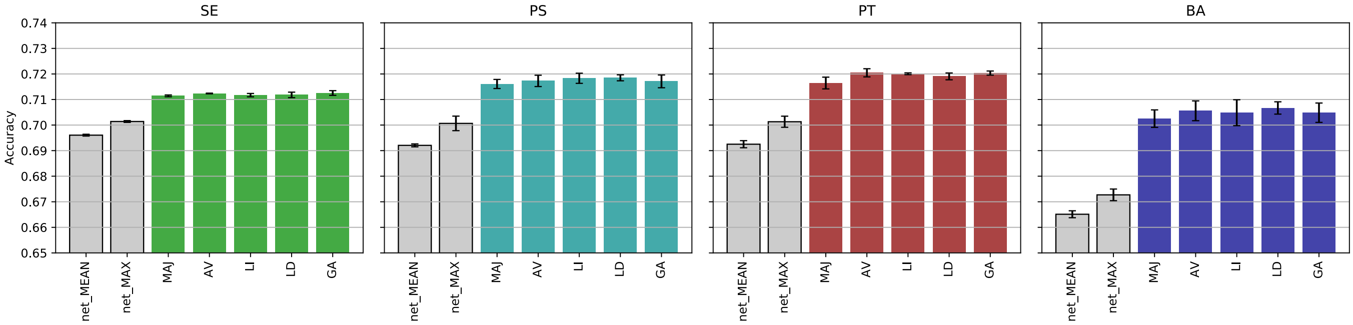

Fig. 6. Bar plots show average results grouped by strategy. In each plot, the first two bars represent the average and the highest classification rates obtained by the single base classifiers, respectively; the other bars represent classification rates obtained by using different aggregation functions. Results are obtained averaging 3 ensembles of 9 networks each. The error bars represent the standard deviation over the three repetitions.

Table 2
Average accuracy in case of base classifiers and ensembles, and ensemble gains for all the different strategies proposed in this work.

Strategy	Average base classifier accuracy		Ensemble accuracy		Ensemble gain	
	Single run	Mean/StdDev	Single run	Mean/StdDev	Single run	Mean/StdDev
SE	69.561 ± 0.451	69.606 ± 0.035	71.245	71.236 ± 0.013	1.684	1.630 ± 0.047
	69.608 ± 0.362		71.245		1.637	
	69.648 ± 0.332		71.218		1.570	
PS	69.128 ± 0.596	69.206 ± 0.057	72.026	71.728 ± 0.220	2.898	2.522 ± 0.271
	69.230 ± 0.489		71.496		2.266	
	69.261 ± 0.363		71.663		2.402	
PT	69.081 ± 0.538	69.252 ± 0.139	72.249	72.044 ± 0.160	3.168	2.793 ± 0.265
	69.252 ± 0.508		71.858		2.606	
	69.422 ± 0.573		72.026		2.604	
BA	66.617 ± 0.392	66.512 ± 0.136	70.577	70.558 ± 0.387	3.960	4.046 ± 0.387
	66.599 ± 0.488		71.023		4.424	
	66.320 ± 0.426		70.075		3.755	

is set to 0.0005, the learning rate schedule is determined by the array of step boundaries [400, 600, 800, 1000, 1200] and the array of learning rate values [0.05, 0.02, 0.01, 0.005, 0.002, 0.001]. The maximum number of epochs is set to 150 and 500 for the CK+ and the SFEW datasets, respectively.

5. Experimental results

The performance metric used in the present work is the accuracy on the reference dataset, i.e. the percentage of correctly classified examples. We evaluated the four selected strategies relying on two measures: the *base classifier accuracy* and the *ensemble accuracy*. The difference between these two values represents the performance gain, obtained thanks to the combination of the base classifiers. For each strategy we performed three repetitions with different seeds in order to assess the stability of the measures. Results concerning single base classifiers and ensembles are summarized in Fig. 6 and in Table 2.

5.1. Accuracy analysis

According to the results, average accuracies for the base classifiers are fairly stable across different runs, independently of the strategy. The highest average accuracy is achieved by SE networks: This could be ascribed to the usage of solely default images without additional pre-processing. Indeed, we observed that the introduction of contrast enhancement leads to a slight performance drop. The SE networks have also the lowest standard deviation of accuracy values, indicating that the introduction of other factors of variation (e.g. pre-training and pre-processing) increases the variability of the trained models. The BA networks achieve the lowest accuracy: The procedure of *sampling with replacement* on the

Table 3
Number of ensembles per ensemble size values obtained by 3 groups of 9 networks for each strategy.

Ensemble size	9	18	27
Number of ensembles	3	3	1

training dataset reduces the total number of different images that each network experiences by a factor of ∼0.63, thus reducing its generalization capability. There is no significant difference between the mean accuracy values of PS and PT networks.

5.1.1. *Comparison of aggregation methods*

As can be noted in Fig. 6, all the aggregation methods employed in this work achieve very close accuracy values, independently of the selected ensemble strategy. Each of the fusion methods yields accuracy values that are significantly higher than those of any base classifier. At least for Pretraining and Bagging Strategy, Average voting (AV) performs better than Majority voting (MAJ). This slight performance drop could be ascribed to the information loss of Majority voting, since it only takes into account the predicted label instead of the posterior class probability. The difference among OWA operators is negligible for all the proposed strategies. Hereafter, for the sake of simplicity, we show and discuss only the results concerning AV.

5.1.2. *Ensemble accuracy*

The SE strategy achieves the lowest ensemble gain: adding preprocessing and pre-training to seed variation helps increase the gain values. The accuracy gains obtained by the PS and PT strategies are similar to each other. Bagging guarantees significantly higher gain values, combining the predictions of networks trained on different subsets of the same dataset (i.e. with the same fingerprint, without any dataset bias issue).

It is straightforward observing that the overall ensemble accuracy depends on the average accuracy achieved by the base classifiers and on the ensemble gain: the highest ensemble accuracy values are achieved by the PS and PT strategies, thanks to the high values of accuracy achieved by the corresponding base classifiers. These strategies achieve comparable values and these values are significantly higher than the ones obtained by the BA and SE strategies. The stability among different measures of ensemble accuracy for the SE strategy is not confirmed for the other strategies, thus suggesting that the variability introduced by the three different seeds for each group leads to slightly different ensembles.

5.1.3. *Increasing the number of base classifiers*

We have also evaluated how the ensemble performance is affected by an increase in the number of base classifiers. We evaluated the Average voting performance for ensembles of 9, 18 and 27 base classifiers. For each strategy, we employed the three ensembles of 9 base classifiers obtained by adopting three different repetitions, as explained above. Thus, per sizes 9, 18 and 27 we measured the accuracy of, respectively, 3 ensembles of 9 networks, 3 ensembles of 18 networks (using different pairs of the ensembles of 9 networks) and 1 ensemble of 27 networks (only one ensemble of 27 networks can be generated by using the three ensembles of 9 networks), as shown in Table 3. Results are shown in Fig. 7.

As expected, the BA strategy has the highest increase in performance (+0.60%) with the increase of the number of networks in the ensemble. Likewise, the accuracy achieved by the PS strategy increases (+0.49%) with the increase of the number of networks in the ensemble. The SE strategy struggles to show any improvement (+0.09%): No significant accuracy improvement is achieved by increasing the number of networks above 9, if the networks are differentiated only by their seeds. Somewhat surprisingly, the accuracy achieved by the PT strategy (-0.24%) seems to be negatively affected by the increase of the ensemble size; the accuracy achieved by the PT strategy is probably hindered by the fact that the additional networks are pre-trained on the same datasets, thus they do not contribute to variability.

Table 4
Number of ensembles per ensemble size values: 6 groups of 9 networks are available for Bagging and Pretraining Strategies.

Ensemble size	9	18	27	36	45	54
Number of ensembles	6	15	20	15	6	1

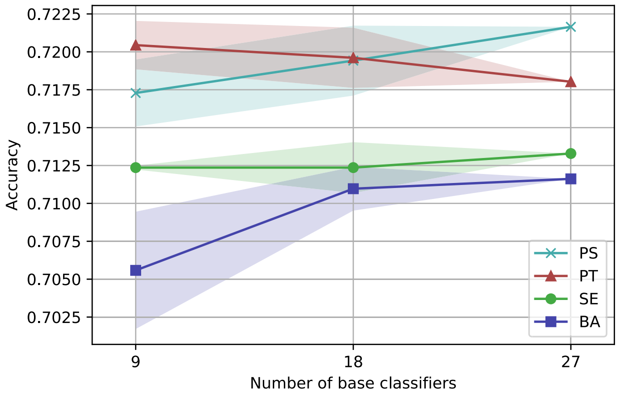

Fig. 7. Average accuracy values versus number of base classifiers in the ensemble (ensemble size up to 27) for all the strategies. The bands represent the standard deviation evaluated on three values for the ensembles with 9 and 18 base classifiers. One single accuracy value is available for the ensemble with 27 base classifiers.

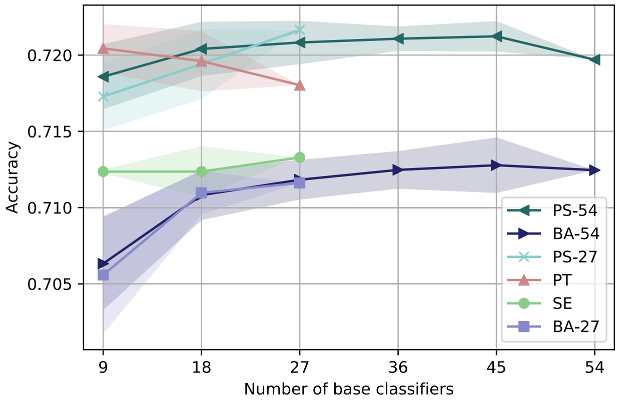

Fig. 8. Average accuracy values of ensembles versus number of base classifiers (ensemble size up to 54) for the BA and PS strategies. The plot is superimposed to the one reported in Fig. 7.

Since the increase in accuracy of the BA and PS strategies is almost linear with respect to the number of base classifiers in the ensemble, we extended the analysis for these two strategies by increasing the ensemble sizes in order to verify whether this trend was confirmed for sizes higher than 27. By training other 3 ensembles with 9 base classifiers, we could rely on several repeated measures for different ensemble sizes: we computed the average values as indicated in Table 4 by using all the possible combinations. Results are presented in Fig. 8. In Fig. 9 we report the first order discrete difference of the accuracy plots in Fig. 8.

As regards the BA strategy, it is worth noticing that, as expected, the performance does not increase linearly with the number of base classifiers. In Fig. 9 it can be observed that the performance gain decreases as the number of base classifiers increases, and eventually becomes negative. As a consequence, the plateau

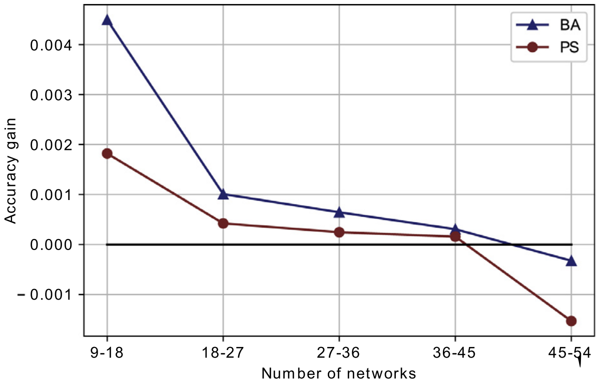

Fig. 9. Difference between average accuracy values of consecutive ensemble size groups in the Bagging and Preprocessing strategies.

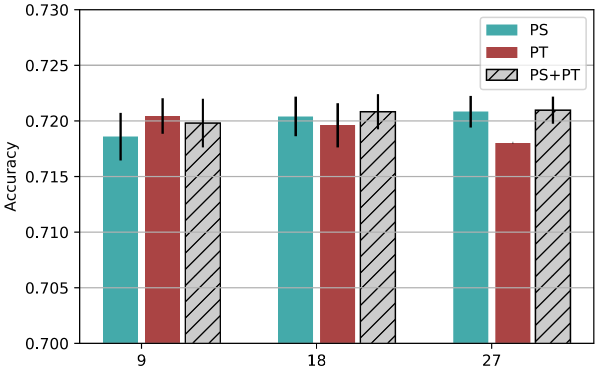

Fig. 10. Average accuracy versus number of base classifiers in the ensemble for the PS and PT strategies and for their combination.

is reached at an accuracy value significantly lower than the other strategies. The performance gain obtained by increasing the ensemble size is lower than the one obtained with BA strategy, reaching negative values. Finally, it is worth underlining that an ensemble strategy that let us achieve good results yet keeping low the ensemble size is certainly preferable, since training from scratch a lot of networks is an extremely time-consuming activity.

5.1.4. Merging the best ensembles

As last experiment, we merged the two best performing ensembles: base classifiers from the PS and PT strategies have been combined in a PS+PT Strategy. The bar plot in Fig. 10 shows that no significant improvement is obtained by using the proposed aggregation. Nevertheless, the average accuracy is more stable with the increase of the number of base classifiers than using the single strategies separately. Thus, we can conclude that the best strategies for generating ensembles of CNNs in the context of FER are the Preprocessing Strategy and Pretraining Strategy. Also, merging the two strategies is not convenient with respect to the delivered average accuracy, but the stability of results is higher with respect to the number of base classifiers.

6. Conclusion

We have presented a comprehensive comparison of four different strategies for the design of an ensemble of CNNs in the context of facial expression recognition. The Seed Strategy simply combines CNNs generated by using different pseudorandom number generator initializations; since the generator affects the behaviour of several CNN components, variability is thus induced. The Preprocessing Strategy employs diverse image preprocessing methods and different seeds. The Pretraining Strategy aims to introduce a factor of variation among the CNNs by pre-training them using different datasets: this leads to different parameter configurations that are used as initial configurations for fine-tuning the networks on the dataset of interest. Finally, the Bagging Strategy introduces a bagging step in the image input pipeline. The comparison between the strategies has been carried out by using also different ensemble aggregation methods. Besides majority voting, a few Ordered Weighted Averaging operators were experimented: results suggest that the simple unweighted average voting is a good choice for the considered case study.

As regards the ensemble design strategies, some conclusions can definitively be drawn: First and foremost, the random components of the network (shuffle of the dataset, dropout, data augmentation, initial distribution of the weights) do not provide enough variability, thus limiting the ensemble performance. This conclusion becomes particularly evident when increasing the number of base classifiers.

The Pretraining strategy and Preprocessing strategy yield similar results, considering both average accuracy of the base classifiers and average accuracy of the ensemble. We have however to consider that the Pretraining Strategy requires additional resources, i.e. additional datasets (SFEW and CK+ datasets in this paper) and additional pre-training time, with no significant increment in performance. Bagging may seem a simpler yet effective alternative, since the relative ensemble guarantees the highest gain respect to the average accuracy of the base classifiers. However, the accuracy values of base classifiers are dramatically affected by the inadequate size of the training set. Therefore, bagging is not an appropriate choice for this kind of application: even increasing the number of base classifiers, the overall accuracy level is lower than the one obtained by employing the other strategies.

We obtained the best absolute test accuracy of 72.249% by using an ensemble of nine networks with the Pretraining strategy. We can conclude that this is the most effective way to build up the ensemble classifier (at the cost of getting information from additional sources). It significantly outperforms the human accuracy, getting to results that are comparable with those of other works evaluated on the same dataset. Some of the related works outlined in Section 1 boosted further the FER2013 test accuracy, but they either considered different settings, namely network trained on different datasets (Zhang et al., 2015b), or introduced other image processing steps, namely face alignment (Kim et al., 2016a) and Scale-Invariant Feature Transform (SIFT) (Connie et al., 2017).

It is worth underlining that our work, unlike other related studies mentioned in Section 1, does not aim to propose completely novel approaches in the field of deep learning for Facial Expression Recognition. Conversely, the novelty of our contribution consists in assessing the performance of various simple ensemble design strategies, shedding light on the factors that can be leveraged to generate diverse and accurate ensembles of neural networks. Furthermore, our broad analysis of ensemble learning focused on aspects not previously considered in the literature: the fusion scheme adopted to combine the output of base classifiers, and the effect of increasing the ensemble size. We believe that such comprehensive analysis can both help practitioners in choosing the ensemble and aggregation strategies, and guide researchers in developing more efficient approaches to exploit ensembles of deep CNNs. The current analysis is restricted to the Facial Expression Recognition task; however, we think that the presented results are quite general for Computer Vision classification tasks that exploit neural networks, since the proposed model and techniques are not specific for the considered task.

Despite the wide scope of the presented investigation on ensemble design strategies, our study obviously has not covered *all* the possible techniques to introduce variability among base classifiers. In future research we will focus on assessing the efficiency deriving from changes/perturbations of the network architecture and from using bigger, albeit non domain-specific, datasets for network pre-training. A further interesting development of this work would consist in introducing more complex image preprocessing steps and hand-crafted features into the design of base classifiers, pursuing state-of-the-art performance on FER benchmark datasets. A limitation for FER system development is represented by the lack of large datasets: leveraging on newly published databases would allow researchers to further improve the performance of FER systems. Finally, this work addressed only *implicit* (or *direct*) approaches to the design of ensembles of neural networks. By introducing other sources of variability, we would be able to rely on a higher number of base classifiers, thus stimulating investigations on selection strategies in *explicit* approaches to properly limit the ensemble size, possibly delivering even more accurate composite classifiers.

Author contributions

Alessandro Renda and Marco Barsacchi shaped the presented approach. The initial ideas have been refined under the coordination of Alessio Bechini and Francesco Marcelloni. Alessandro Renda implemented the required software and performed the experimental tests, according to the indications decided with all the other authors. All the authors contributed to writing the final manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was been partially supported by the University of Pisa under grant PRA_2017 “IoT e Big Data: Metodologie e tecnologie per la raccolta e l’elaborazione di grosse moli di dati.”

References

Alajlan, N., Bazi, Y., AlHichri, H. S., Melgani, F., & Yager, R. R. (2013). Using OWA fusion operators for the classification of hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6*(2), 602–614. doi:10.1109/JSTARS.2013.2240437.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning, 2*(1), 1–127. doi:10.1561/2200000006. Also published as a book. Now Publishers, 2009.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140. doi:10.1023/A:1018054314350.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: A survey and categorisation. *Information Fusion, 6*(1), 5–20. doi:10.1016/j.inffus.2004.04.004. Diversity in Multiple Classifier Systems.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., & LeCun, Y. (2015). The loss surfaces of multilayer networks. In *Proc. of the 18th conf. on artificial intelligence and statistics*. In *Proceedings of Machine Learning Research: 38* (pp. 192–204).

Connie, T., Al-Shabi, M., Cheah, W. P., & Goh, M. (2017). Facial expression recognition using a hybrid CNN–SIFT aggregator. In *International workshop on multi-disciplinary trends in artificial intelligence* (pp. 139–149). Springer. doi:10.1007/978-3-319-69456-6_12.

Dhall, A., Goecke, R., Joshi, J., Sikka, K., & Gedeon, T. (2014). Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *Proceedings of the 16th international conference on multimodal interaction* (pp. 461–466). ACM.

Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2012). Collecting large, richly annotated facial-expression databases from movies. *IEEE MultiMedia, 19*(3), 34–41. doi:10.1109/MMUL.2012.26.

Dornaika, F., Moujahid, A., & Raducanu, B. (2013). Facial expression recognition using tracked facial actions: Classifier performance analysis. *Engineering Applications of Artificial Intelligence, 26*(1), 467–477. doi:10.1016/j.engappai.2012.09.002.

Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology, 17*(2), 124. doi:10.1037/h0030377.

Fodor, J., Marichal, J.-L., & Roubens, M. (1995). Characterization of the ordered weighted averaging operators. *IEEE Transactions on Fuzzy Systems, 3*(2), 236–240. doi:10.1109/91.388176.

Gan, Y., Chen, J., & Xu, L. (2019). Facial expression recognition boosted by soft label with a diverse ensemble. *Pattern Recognition Letters, 125*, 105–112. doi:10.1016/j.patrec.2019.04.002.

Giacinto, G., & Roli, F. (2001). Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing, 19*(9), 699–707. doi:10.1016/S0262-8856(01)00045-2.

Gonzalez, R. C., & Woods, R. E. (2006). *Digital image processing (3rd edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. http://www.deeplearningbook.org

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., … Bengio, Y. (2015). Challenges in representation learning: A report on three machine learning contests. *Neural Networks, 64*, 59–63. doi:10.1016/j.neunet.2014.09.005. Special Issue on Deep Learning of Representations.

Gross, R., & Brajovic, V. (2003). An image preprocessing algorithm for illumination invariant face recognition. In *Proc. of the 4th int’l conf. on audio- and video-based biometric person authentication* (pp. 10–18). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/3-540-44887-X_2.

Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-PIE. *Image and Vision Computing, 28*(5), 807–813. doi:10.1016/j.imavis.2009.08.002.

Han, D., Liu, Q., & Fan, W. (2018). A new image classification method using CNN transfer learning and web data augmentation. *Expert Systems with Applications, 95*, 43–56. doi:10.1016/j.eswa.2017.11.028.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proc. of the IEEE conf. on computer vision and pattern recognition* (pp. 770–778). doi:10.1109/CVPR.2016.90.

Hertel, L., Barth, E., Kster, T., & Martinetz, T. (2015). Deep convolutional neural networks as generic feature extractors. In *2015 international joint conference on neural networks (IJCNN)* (pp. 1–4). doi:10.1109/IJCNN.2015.7280683.

Hussain, A., Khan, M. S., Nazir, M., & Iqbal, M. A. (2012). Survey of various feature extraction and classification techniques for facial expression recognition. In *Proc. of the 11th wseas int’l conf. on electronics, hardware, wireless and optical communications, and proc. of the 11th wseas int’l conf. on signal processing, robotics and automation, and proc. of the 4th wseas int’l conf. on nanotechnology* (pp. 138–142). Stevens Point, Wisconsin, USA: WSEAS.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift* arXiv:1502.03167.

Jaitly, N., & Hinton, G. E. (2013). Vocal tract length perturbation (VTLP) improves speech recognition. In *Proc. icml workshop on deep learning for audio, speech and language: 117*.

Ju, C., Bibaut, A., & van der Laan, M. J. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 1–19. doi:10.1080/02664763.2018.1441383.

Khorrami, P., Paine, T., & Huang, T. (2015). Do deep neural networks learn facial action units when doing expression recognition? In *Proc. of the IEEE int.l conf. on computer vision workshops* (pp. 19–27). doi:10.1109/ICCVW.2015.12.

Kim, B.-K., Dong, S.-Y., Roh, J., Kim, G., & Lee, S.-Y. (2016a). Fusing aligned and non-aligned face information for automatic affect recognition in the wild: A deep learning approach. In *Proc. of the ieee conf. on computer vision and pattern recognition workshops* (pp. 1499–1508). doi:10.1109/CVPRW.2016.187.

Kim, B.-K., Roh, J., Dong, S.-Y., & Lee, S.-Y. (2016b). Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. *Journal on Multimodal User Interfaces, 10*(2), 173–189. doi:10.1007/s12193-015-0209-0.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105). Curran Associates, Inc.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. doi:10.1038/nature14539.

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (1998). Efficient backprop. In *Neural networks: Tricks of the trade* (pp. 9–50). Springer. doi:10.1007/3-540-49430-8_2.

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. In *Proc. of 2010 ieee int’l symposium on circuits and systems (iscas)* (pp. 253–256). IEEE. doi:10.1109/ISCAS.2010.5537907.

Liang, D., Liang, H., Yu, Z., & Zhang, Y. (2019). Deep convolutional bilstm fusion network for facial expression recognition. *The Visual Computer*. doi:10.1007/s00371-019-01636-3.

Liu, H., & Zhang, L. (2019). Advancing ensemble learning performance through data transformation and classifiers fusion in granular computing context. *Expert Systems with Applications, 131*, 20–29. doi:10.1016/j.eswa.2019.04.051.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The extended Cohn–Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *Proc. of 2010 ieee computer society conf. on computer vision and pattern recognition workshops (cvprw)* (pp. 94–101). IEEE. doi:10.1109/CVPRW.2010.5543262.

Lyons, M. J., Akamatsu, S., Kamachi, M., & Gyoba, J. (1998). Coding facial expressions with Gabor wavelets. In *Proc. of 3rd int’l conf. on automatic face and gesture recognition* (pp. 200–205). doi:10.1109/AFGR.1998.670949.

Mahersia, H., & Hamrouni, K. (2015). Using multiple steerable filters and Bayesian regularization for facial expression recognition. *Engineering Applications of Artificial Intelligence, 38*, 190–202. doi:10.1016/j.engappai.2014.11.002.

Martinez, B., & Valstar, M. F. (2016). Advances, challenges, and opportunities in automatic facial expression recognition. In *Advances in face detection and facial image analysis* (pp. 63–100). Springer.

Ponti Jr, M. P. (2011). Combining classifiers: From the creation of ensembles to the decision fusion. In *24th sibgrapi conf. on graphics, patterns and images tutorials (sibgrapi-t)* (pp. 1–10). IEEE. doi:10.1109/SIBGRAPI-T.2011.9.

Pramerdorfer, C., & Kampel, M. (2016). *Facial expression recognition using convolutional neural networks: State of the art* arXiv:1612.02903.

Sariyanidi, E., Gunes, H., & Cavallaro, A. (2015). Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*(6), 1113–1133. doi:10.1109/TPAMI.2014.2366127.

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition* arXiv:1409.1556.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929–1958.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proc. of the ieee conf. on computer vision and pattern recognition* (pp. 2818–2826). doi:10.1109/CVPR.2016.308.

Tang, Y. (2013). *Deep learning using linear support vector machines* arXiv:1306.0239.

Tie, Y., & Guan, L. (2013). Automatic landmark point detection and tracking for human facial expressions. *EURASIP Journal on Image and Video Processing, 2013*(1), 8. doi:10.1186/1687-5281-2013-8.

Valstar, M., & Pantic, M. (2010). Induced disgust, happiness and surprise: An addition to the MMI facial expression database. In *Proc. 3rd int.l workshop on emotion (satellite of lrec): Corpora for research on emotion and affect* (p. 65).

Wen, G., Hou, Z., Li, H., Li, D., Jiang, L., & Xun, E. (2017). Ensemble of deep neural networks with probability-based fusion for facial expression recognition. *Cognitive Computation, 9*(5), 597–610. doi:10.1007/s12559-017-9472-6.

Xie, S., Hu, H., & Wu, Y. (2019). Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition. *Pattern Recognition, 92*, 177–191. doi:10.1016/j.patcog.2019.03.019.

Yager, R. R. (1993). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. In D. Dubois, H. Prade, & R. R. Yager (Eds.), *Readings in fuzzy sets for intelligent systems* (pp. 80–87). Morgan Kaufmann. doi:10.1016/B978-1-4832-1450-4.50011-0.

Yager, R. R. (2007). Using stress functions to obtain OWA operators. *IEEE Transactions on Fuzzy Systems, 15*(6), 1122–1129. doi:10.1109/TFUZZ.2006.890686.

Zhang, X., Mahoor, M. H., & Mavadati, S. M. (2015a). Facial expression recognition using l_p-norm MKL multiclass-SVM. *Machine Vision and Applications, 26*(4), 467–483. doi:10.1007/s00138-015-0677-y.

Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A., … Girard, J. M. (2014). BP4D-Spontaneous: A high-resolution spontaneous 3D dynamic facial expression database. *Image and Vision Computing, 32*(10), 692–706. doi:10.1016/j.imavis.2014.06.002.

Zhang, Z., Girard, J. M., Wu, Y., Zhang, X., Liu, P., Ciftci, U., … Yin, L. (2016). Multimodal spontaneous emotion corpus for human behavior analysis. In *Proc. of the IEEE conf. on computer vision and pattern recognition* (pp. 3438–3446). doi:10.1109/CVPR.2016.374.

Zhang, Z., Luo, P., Loy, C.-C., & Tang, X. (2015b). Learning social relation traits from face images. In *Proc. of the IEEE int'l conf. on computer vision* (pp. 3631–3639). doi:10.1109/ICCV.2015.414.