Emotion Recognition
Neural Networks for the analysis of EEG signals

[redacted]

Abstract—**Electroencephalograms are useful signals used in human data machine learning elaborations for the analysis and predictions of different aspects. One of them is the emotion recognition: it is the capability of a model to predict, from an EEG signal, if an emotion, in this case Arousal or Valence, has been proved. This analysis could be useful for many fields, e.g. commercial or medical. In this paper, different features extracted from signals - data mapping, statistical features and in frequency-domain signals - and different neural networks - deep, convolutional, recurrent, residual and autoencoder - have been tested to predict emotions. The target is to find which combination allows to obtain an higher accuracy when one or two labels are considered. The best performances have been reached with convolutional neural networks using as input the entire signals coming from single EEG channels: accuracies of 97% for one label and 89% for two labels are obtained. Interesting results has been achieved also applying Kernel Principal Component Analysis to windowed data or extracting statistical characteristics from them. This work can represent a state of art for subsequent works: type of data and network that lead to be performances can be chosen for the implementation of models for this task since many other network combinations could be tested starting from these.**

Index Terms—**Supervised Learning, Feature Extraction, Neural Networks, Deep Learning, Electroencephalogram.**

I. Introduction

In last years, the need of analyzing human data is an increasing challenging task for different purposes, e.g. health monitoring systems to analyze and predict possible diseases. Emotion recognition could be consider as one of the most challenging machine learning problems in this field since the labels are subjective and, as consequence, difficult to predict. The emotions can be recognized from many visual aspects or body signals: many studies reveal that the electroencephalogram (EEG) signal is one of the most meaningful aspects. The standard way to obtain and analyze human data exploits the Brain Computer Interface, needed for the transmission of these data from the sensors to the computer; machine learning procedures are then needed to learn supervised and unsupervised models for classification and analysis. The application of these types of studies allows also to better understand the correlation between the EEG signals registered during the stimulus and the emotion proved, suggesting in this way further important information about the human brain behaviour.

To build models for emotion prediction, the DEAP dataset [1] can be used: EEG signals of people looking at music videos are registered from 32 EEG channels. Each participant assigned a label in each trial: this label reports the 'Arousal' and 'Valence' emotions felt while watching the video. Valence describes the level of positivity or negativity of the subject while arousal measures the level of excitement felt. According to the dimension approach for the emotion classification (one of many present in literature), every emotion can be classified according to these two labels and find a good classifier for them can be a general result for the EEG analysis.

Different articles are present in literature with the purpose of build accurate neural networks to predict the label associated to EEG signals. In [2] different data preprocessing, mainly related to the statistical analysis of the signal, are proposed as input of a simple convolutional neural network and of a combination of convolutional neural network, autoencoder and deep neural network. In [3] time and frequency domain data (and combination of them) are used to predict the label with a deep convolutional neural network and the results are compared with standard classification algorithms. In [4] the performances of a deep neural network and a convolutional neural network are compared starting from the statistical characteristics of the data.

From the papers cited above the reader can notice how the neural networks well works with respect to other classifications methods and even if they demonstrate that very complex models are needed to find good classifiers for this task, they do not reach in general perfect generalization for the dataset. By consequence, it is still an open problem to be solved.

For these reasons, in this work a wide proposal of neural networks are suggested to evaluate and compare their performances, in order to find the best starting point to build an accurate model for this task. The purpose of this paper is to study not only the most appropriate neural network that permits to correctly classify each sample in terms of high or low arousal and valence, but also different types of data preprocessings. In fact, these procedures may extract important signal features and lead to more accurate classifications. The elaborations could also reduce the dimensionality of data feed as input to the network and, by consequence, achieve faster trainings and predictions of the model, reducing the model complexity and so trying to avoid overfitting.

In synthesis, this paper wants to analyze:

- Different data preprocessing in order to find the best way to represent the most important information needed for the classification and for the dimensionality reduction of the dataset.
- Different structures of neural networks in order to find the best model structure able to catch the hidden information

[redacted]

and predict emotions with high precision.
- The comparison of the results obtained with the same features and models for a single label and for both the labels together.
- The comparison between the prediction using directly EEG signals with respect to the ones with features extracted from them.

For these purposes, this report is structured as follows: In Section II the state of art is described; in Section III the main work pipeline is presented, while in Section IV the data used and the features extracted. In Section V the compared models are described. The performance evaluation of the combinations of models and processing techniques are proposed in Section VI, while conclusions are reported in Section VII.

II. Related Work

Some recent papers exposes some solutions for this task with specific properties or neural networks. Many studies proposed a comparison between performances of supervised and unsupervised learning techniques: they demonstrated that neural networks are the main procedure able to detect recognizable patterns useful for the emotion classification.

In [2], the authors extracted from DEAP dataset the raw, the normalized and in frequency domain data and used them and their combination to compare the results of different classification procedures. In particular, they used bagging tree, support vector machine, linear discriminant analysis, Bayesian linear discriminant analysis models and deep convolutional neural networks. In terms of average AUC after the training, the deep CNN model achieved the best performances in both temporal and frequency domains with respect to all the others models. Considering the valence label, raw data and normalized data performed with AUC values of around 0.6, while frequency data increased the performances, reaching 0.96, this time for the bagging tree method. Lastly, the two combined data had values between 0.65 and 1, with best performances with deep convolutional neural networks, resulting in conclusion the more appropriate combination of feature and model. The combined features had in almost all cases a much higher value of AUC with respect than those on all single features. Same results were almost achieved for arousal, demonstrating that there is no important difference in predicting these two emotions. In this study, the higher capability of the emotion prediction of the neural networks emerges with respect to the other procedures. However, it is not analyzed the case of both the labels and only few type of data processing are considered in order to find the best for this type of classification.

In [5], a Long-Short Term Memory (LSTM) mechanisms is used to learn features from EEG signals and these are subsequently feed to a dense layer classifier to determine low/high arousal, valence, and liking. In this case, accuracies of around 85% are obtained for both arousal and valence. This recurrent procedure permits a good classification for a single label. In this case, no previous feature extraction is performed and this procedure could increase the network performances.

In [3], instead, a particular neural network composed by a convolutional network, a sparse autoencoder and a deep neural network is trained for the classification. The convolutional network is useful to extract data features, the subsequent autoencoder is used to remove redundancy and finally the deep network is used for a good classification. This network is evaluated with respect to a deep convolutional neural network for features extracted with Principal Component Analysis, Pearson correlation coefficients and statistical characteristics from data divided into time windows of 8 and 12 seconds. In this case, the proposed network had always an higher level of accuracy with respect to the standard convolution. With a time windows of 8 seconds, the CNN has maximum accuracy with Pearson correlation coefficients of around 80%, while the proposed network has accuracies of 89.49% and 92.86%, respectively for valence and arousal. For time windows of 12 seconds, all the accuracy values decreases of about 5% but maintaining the same relationship, demonstrating that collecting emotion information is more difficult when the stimulation time increases. In this work, a very particular neural network is tested and quite good results have been obtained but no other types of network are tested. In addiction, also in this case the prediction of both labels at the same time is not implemented.

In [4], a deep neural network and a convolutional neural network are used for the classification starting from the preprocessed data and its statistical characteristics partitioned into windows of 2 seconds. They obtained accuracies of around 75% for valence and arousal with DNN, accuracy of 81% for valence and 73% for arousal with CNN. During the experiments, they noticed that the CNN seemed to achieve higher accuracy but a much higher range was present between subject classification accuracies. This work wants to introduce the state-of-art for this task but few features and types of neural networks are proposed. In this paper, this work has been expanded introducing more of them.

With respect to all these studies, in this paper a more wide comparison between possible signals derivations and feature extraction techniques and neural network models is presented, for both single label and the double labels.

III. Processing Pipeline

Starting from the DEAP dataset that contains multiple EEG signals obtained from 32 channels of 32 participants, an initial pre-processing has been performed. Initially, data are divided thanks to pass band filters in the four bands included in the [4, 45] Hz range of standard frequencies of EEG signals. Each band is then standardized in order to have no bias due to possible different intensities of signals. Two main types of signal are then taken into account: considering each participant and each sample, the first type proposed is the entire EEG coming from a single channel while the second one is derived from time-windows of the EEG signals considering together the 32 channels. From this second group, some data elaborations are done to extract the related features: these procedures are Principal Component Analysis, Kernel Principal Component

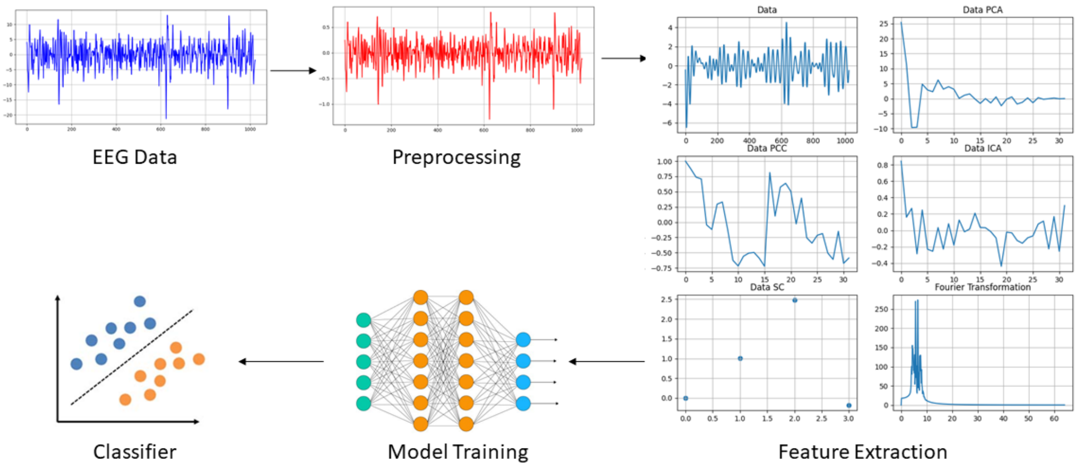

Fig. 1: Pipeline. Starting from EEG signals, they are standardized and divided with pass-band filters, then different features are extracted. On the right, features extracted for one time-window and a single channel are plotted. At the end, a neural network is trained to build a classifier.

Analysis, Independent Component Analysis, Pearson Correlation Coefficients, Independent Component Analysis, Statistical Characteristics and Fourier Transformation. Some of these features are extracted to find some type of data correlation between the different channels, while other information are extracted separately for each channel inside the data window.

For all these datasets, different types of neural networks are fit to evaluate their performances: deep neural network, convolutional neural network, dilated convolutional neural network, locally connected network, recurrent neural network and a network derived from an autoencoder. The performances are then evaluated in terms of accuracy on the test set.

All this process must be repeated three times: one for each single label and the last one for both the labels together.

IV. Signals and Features

The signals contained in the DEAP dataset [1] are divided for the 32 participants of the trial. The 40 EEG signals of each participant were recorded while showing him 40 videos: they were recorded by 32 electrodes and 8 body sensors.

In the ‘participant_ratings.csv’ file in the dataset, each video is evaluated by each participant with different values in terms of “Valence”, “Arousal”, “Dominance”, “Liking” and “Familiarity”. Since the combination of Valence and Arousal can represent the entire set of emotions, in this work only these two are considered. Since every sample is associated with Arousal and Valence labels with integers between 0 and 9 and the purpose is to classify only high or low levels of the two emotions, a label transformation is performed: for both of them, emotion values from 0 to 4 are converted into a 0 label while values greater than 4 have label equal to 1.

Since only the first 32 signals were derived from EEG sensors, only those are maintained. Each signal is composed of 63 seconds. The first three seconds are a pre-trial baseline, removed before the computation. For the signals, the sampling rate for the entire experiment was of 128 Hz.

An important aspect that must be taken into consideration in the EEG analysis is that these signals very often contain artifacts due principally to the eyes movements. Biological studies have demonstrated that EEG signals are characterized from frequency between 4 and 45 Hz: in this work, the other frequencies are discarded, with the purpose of avoiding possible artifacts at different frequencies. In particular, the four most important frequency bands are the following: δ between 4 and 8 Hz, θ between 8 and 14 Hz, α between 14 and 30 Hz and β between 30 and 45 Hz. To allow the extraction of features or important patterns within or between bands during the learning process, each signal is divided into its four bands. In the subsequent analysis, the four bands will always left separated and the computation will be performed independently for each of them.

After this first initial data creation, for each band and separately for each channel data are standardized in order to obtain everywhere mean zero and standard deviation one. This step is necessary for the subsequent feature extractions and to avoid possible biases in the learning.

Subsequently this initial preprocessing, two main types of signal are considered: the first contains entire signals of 60 seconds (sampled with rate 128 Hz, so 7680 total values) coming from a single channel, from each video and each participant. For this reason, this dataset contains 32 (participant) x 40 (video) x 32 (channels) samples, each one with 4 (bands) x 7680 values. The values are then reshaped, obtaining matrices with on the rows the 60 seconds and on the columns 128 values sampled in the correspondent second. The second dataset contains signals coming from the same participant and same video and obtained thanks to time-windows considering all the EEG channels. To have overlapping, not too small time-windows (to have enough information contained) but neither too large (since [3] demonstrated that smaller time-windows were more accurate in the classification), in this work windows of 8 seconds are consider with shifts of 5 seconds between them. Being the signals of 60 seconds, 14 windows can be extracted, each one containing 8 (seconds) x 128 (sampling-rate). The dataset contains so 40 (participant) x 32 (video) x 14 (windows) samples. Each sample has a shape of 4 (bands) x 32 (channels) x 1024.

From these seconds time-windows, the features IV-A, IV-B, IV-C, IV-D are derived to identify correlations between the channels in the time intervals. Procedures IV-E and IV-F, instead, reveal characteristics within the same channel. The number of samples in the datasets remains constant since it derives from the number of windows generated from each signal, but the size of the sample will now change and it is reported for each data transformation.

A. *Principal Component Analysis*

The Principal Component Analysis is a statistical method for the dimensionality reduction: in an unsupervised manner, it extracts the orthogonal directions that maximize the variance of data. The optimized procedure consists in a linear decomposition into eigenvectors of the data matrix passed as input. With this analysis, the first 32 components of maximum variance are extracted from data for each pass-band filtered signal and these data are then mapped into the space defined by these new components. Each sample is composed by 4 (one for each band) matrices, each of shape 32 x 32.

B. *Kernel Principal Component Analysis*

The Kernel Principal Component Analysis is a variant of the standard PCA that permits to deal with more complex data patterns, which would not be analysable with simple linear transformations. In fact, this technique applies to the input matrix a non-linear transformation and then extracts the principal components in the standard way. Since this operation could be very time consuming, the kernel trick can be applied to reduce the complexity. Given a non-linear data transformation $\phi(\mathbf{x})$, the kernel function is by definition $\kappa(\mathbf{x}) = \phi(\mathbf{x})\phi^T(\mathbf{x})$ and already includes both the operations, with the possibility to directly derive the eigenvectors for the principal components. In this study, a Radial-basis (rbf) kernel function has been applied for the extraction of the most significant 32 components and the subsequent data mapping. As in the previous case, the dataset has shape 4 x 32 x 32.

C. *Independent Component Analysis*

Similarly to previous ones, the Independent Components Analysis is a technique that allows to derive the inner components of a signal: assuming data to be linear unknown mixtures of some unknown latent variables, considered non-gaussian and mutually independent, this technique tries to extract these latent variables, called independent components. Due to its very costly computation, the optimized FastIca algorithm has been used for this optimization problem. Some studies assert that this type of analysis on EEG signals permits to find more easily artifacts: it is usually used setting a certain threshold and eliminating the values below this value; in this work, all the components extracted are maintained in order to evaluate if the neural networks are able to learn the correct classification during their training. To perform comparisons with the previous two methods too, the 32 independent principal components are derived and used for the data transformation in the new space. The obtained dataset is so composed by 4 matrices, each of shape 32 x 32.

D. *Pearson Correlation Coefficients*

The Pearson Correlation Coefficient of two random variables X and Y is defined as:

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y},$$

where σ_X and σ_Y are the covariances of the respective variables, while σ_{XY} is the covariance between X and Y. This value, that is in $[-1, 1]$ range, represents if the two variables are directly correlated ($\rho = 1$), not correlated ($\rho = 0$) or inversely correlated ($\rho = -1$). These coefficients are calculated for each data window considering the distribution generated by each channel, in order to find the correlation between them. In this case, so, 4 matrices of 32 x 32 are obtained for each sample.

E. *Statistical Characteristics*

Statistical Characteristics are then extracted in every data window and for every channel. In particular, considering separately the four bands, four characteristics are retained for the data distributions generated by each channel:

1) mean;
2) variance, the variability of data;
3) skewness, the measure of the asymmetry of the data distribution around its mean. It is represented by a value between -1 and 1: equal to zero presents a normally distributed variable, if equal to -1 the distribution has a more important tail on the left side, while if equal to 1 the tail is principally on the right;
4) kurtosis, the measure of the tails size of the data distribution. It is represented by a real value, usually compared with the kurtosis of any normal distribution equal to 3.

In this way, 4 matrices of shape 32 x 4 are obtained and used as input feature for the learning.

F. *Fast Fourier Transformation*

At the end, the Fast Fourier Transformation is a method that allows to convert the time-domain samples in frequency-domain values, where the signal is seen as a linear combination of complex sinusoidal functions. In this case, a real discrete Fourier Transformation is applied separately to each channel of the time-windowed signals, maintaining then its absolute value. The coefficients extracted are 513 and by consequence, the matrices obtained has shape 4 x 32 x 513.

All the datasets are then divided into training, validation and test set: the training set used during training phase contains 70% of the data, the validation set used after each epoch for the training validation contains 20% of the data, while test set, used at the end of the training for the accuracy analysis, contains 10%. In particular, for the signal dataset, 28675 samples are in the training set, 8601 in the validation set and 3687 in the test set; for the datasets derived from time-windowed data, 12544 samples are for training, 3763 for validation and 1613 in the test set.

V. Learning Framework

To all the datasets described before, the following neural networks are trained. Some hyperparameters are common for all the trainings and here described.

The training happens with a mini-batch approach, with batch size of 64 samples. Trainings were performed repeatedly for 20 epochs, to allow a repeated learning procedure in order to obtain a more accurate learning. The training is performed over the entire training set and after each of them, loss and accuracy are tested even for the validation set, to check the obtained model performance. An early stop callback is used to stop in advance the training if the validation loss does not decrease for five consecutive epochs. In addition, during the execution of the learning procedure, ”Adam” optimizer has been used: this Keras built-in method arranges dynamically and differently for every parameter the learning rate, permitting a faster and precise learning.

The labels to be predicted are binary and quite balanced for training, validation and test sets: both arousal and valence are 0 for about 43% and 1 for 57%. Since the labels are binary, the binary cross entropy can be used as loss function and since they are balance, the accuracy is selected as metric.

At the end of the training, the classification capability is evaluated with the test sets with different scores: after data prediction, accuracy, precision, recall, Fscore and AUC are calculated. For each label, in case of the two labels classification, the ROC curve is also printed. In this way, a complete analysis of the network over the test set is allowed.

In the following subsections, the implemented neural networks are presented. All the output layers contain one unit in case of a single label or two units in case of double labels. For this output layer, the sigmoid activation function is always used, since it is the one that better describe the situation of prediction of the labels: they can assume only binary values and, when both labels are considered, they are not mutually exclusive. In figure 2 the final models are reported.

A. Deep Neural Network

A first deep neural network is implemented: it is composed by the input layer, the flatten of the input in one dimension and four fully connected layers. These hidden layers have respectively 512, 256, 128 and 64 units and the relu function is chosen as activation to try to avoid the vanishing gradient problem during the training. In addition, a L1 regularizer defined as

$$\lambda \cdot \sum_i w_i,$$

where w_i are the weights of the network and used with coefficient $\lambda = 10^{-4}$, is added to the cost function of the fully connected layers during the training. It allows to reduce the model complexity, that could lead to the overfitting of the training set. After this deep structure, the output layer is added for the label prediction.

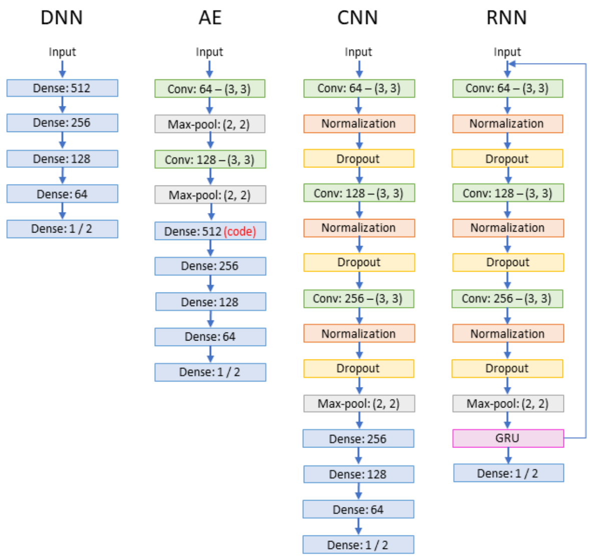

Fig. 2: Different models proposed

B. Autoencoder

The autoencoder is a neural network composed by an encoder and a decoder: the encoder encodes the input data into a code of a defined size. The code is able to represent the main characteristics of data in order to rebuild it with the decoder application. In this work, a convolutional autoencoder has been implemented: in the encoder part, three convolutional 2D layers with respectively 64, 128 and 256 filters and kernel size of (3, 3) are followed by 2D max pooling with size (2, 2). After that, a code of size 512 is extracted for the signal representation. For the decoder part, a first initial dense layer (of size depending on the input dataset) is followed by three transposed convolutional 2D layers with 128, 64 and 4 filters of size (3, 3). Since for the autoencoder input and output are equal to the whole data, it is trained trying to minimize the Mean Squared Error as loss function.

To build the classifier, keeping fixed the encoder weights, a network with the encoder and a deep neural network is used: three layers with 256, 128 and 64 filters of size (3, 3) are added after the code extracted and allow to build the fully-connected network unto the final output layer.

C. Convolutional Neural Network

Another possibility of neural network implementation is the 2D convolutional model. This network is implemented with three 2D standard convolutional layers, respectively with 64, 128 and 256 filters. Every filter of the layers has size of (3, 3) and it is applied with a stride of (1, 1) - also to permit a comparison with the subsequent networks. The values of the kernel size are decided in order to have a trade-off between a not too slow learning (that could be obtained for too big kernel sizes) and a good representation of the relationships among values. In each layer, the padding parameter is set to *same* in order to keep the output size: on the contrary, smallest datasets as the one derived with statistical characteristics would not permit the consecutively application of this operation. After each convolutional layer, three other layers are applied: a batch normalization layer permits data rescaling that allows to obtain a faster and more robust result, an activation layer that applies the relu function to every node and a dropout layer to delete 20% of the values present at the previous layer output and preventing in this way overfitting to happen. After this convolutional block, a 2D max-pooling layer is applied with a pool size of (2, 2) and at the end, after the flattening of the last hidden layer output, the output layer is applied.

After this initial training that permits the network to learn to extract information to classify the input data with convolutional operations, there is the need to build a more accurate classifier for the task. To do this, the last output layer is removed and a deep neural network, as the one described in section V-A but without the first dense layer of 512 units, is added. The weights of the first convolutional part of the network are set as not-learnable, so are kept fixed to maintain the extracted features, while the weights of the added DNN are learnt in order to build a deep classifier.

D. Dilated Convolutional Neural Network

This third neural network is much equivalent to the one described above in terms of layers applied: 2D convolution, batch normalization, activation with relu function and dropout layers, followed by max-pooling and output layers. A difference has been introduced in the application of the 2D convolutional layers, because every convolutional operation is performed with a kernel size of (3, 3) and with a dilation rate of (2, 2). With respect to the standard convolutional layer (equivalent to a dilated layer with dilation rate of (1, 1)), this operation could be able to find more interesting properties exploring a wider combinations of values thanks to the dilation: as a matter of fact, the dilation permits to widen the kernel and with a dilation rate of ℓ, between one element considered by the kernel and the subsequent there are $\ell - 1$ not considered elements. An example of this operation and its comparison with the normal operation is reported in figure 3. With this type of operation, the convolutional layer is applicable only with strides of (1, 1).

After the training of this network, also in this case the output layer is removed and the deep neural network with three hidden layers (with respectively 256, 128 and 64 units, relu activation function and L1 regularizer) is added and trained: in this way, it is possible to achieve a good feature extractor with the convolutional block and a good classifier with the following deep network.

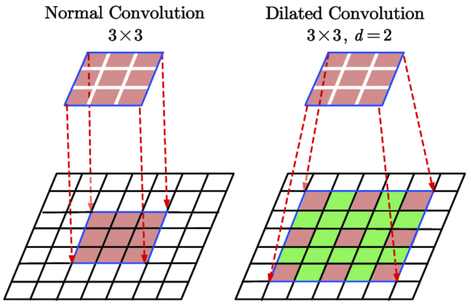

Fig. 3: Convolutional operation and dilated convolutional operation

E. Locally Connected Convolutional Neural Network

A further variant of the standard convolutional layer is represented by the Locally Connected layer. While in the standard convolutional layer the same filter (in terms of weights) is applied to every pixel in the input, in the locally connected layer a different filter is applied to each pixel. With this method, it is possible to find different patterns inside the same data matrix and, in every section of the 2D input, is possible to extract different features. This operation brings to a very high number of parameter to be learnt during the training procedure: this aspect leads its implementation to be a very hard task and, by consequence, still not complete in Keras module. For this reason, the possibility to have the parameter padding equal to *same* is not yet implemented and only *valid* is possible. Caused by this limitation, it has not been possible to apply this type of network to the Statistical Characteristics dataset because of its too low starting dimension that cannot apply neither two or three convolutional operation with valid padding. For the high amount of memory required for the computation, it has not been possible to train this type of model also for the biggest datasets, the ones derived from Signal, Window and Fast Fourier Transformation.

The neural network has been implemented with three locally connected layers respectively with 64, 128 and 256 layers, all with kernel size of (3, 3) and padding *valid*. As in the case of convolutional network, each layer is followed by a batch normalization layer, an activation layer with relu function and a dropout layer with drop rate of 0.2. After the convolutional block, also in this case before the flatting of the output and the output layer, a 2D max-pooling layer is applied.

After the training of this network the last layer is removed and keeping fixed the weights obtained, a Deep Neural Network (of 256, 128 and 64 units, relu activation function and L1 regularizer) is added and trained for the classification task.

F. Recurrent Neural Network

A recurrent neural network is implemented: after the two first hidden layers of the convolutional neural network (64 and 128 filters, with kernel size of (3, 3)), and a reshape

	Arousal				
	DNN	**AE**	**CNN**	**CNNd**	**CNNl**
Signal	0.9406	0.9410	0.9699	0.9473	x
Window	0.5759	0.5793	0.6216	0.6028	x
PCA	0.8122	0.8134	0.8562	0.8717	0.8047
KPCA	0.7006	0.8524	0.8748	0.7756	0.8314
ICA	0.5759	0.6183	0.6072	0.5760	0.5830
PCC	0.6826	0.8326	0.8630	0.8692	0.7706
FFT	0.5759	0.8153	0.8741	0.8496	x
SC	0.7198	0.8047	0.8834	0.8828	x

TABLE 1: Accuracy for Arousal label

	Both Labels				
	DNN	**AE**	**CNN**	**CNNd**	**CNNl**
Signal	0.8644	0.8673	0.8894	0.8398	x
Window	0.3428	0.5346	0.5822	0.6144	x
PCA	0.6578	0.7068	0.7223	0.6417	0.6235
KPCA	0.6708	0.7626	0.7630	0.7358	0.6973
ICA	0.3428	0.5027	0.5983	0.6028	0.5674
PCC	0.3931	0.6792	0.7402	0.7164	0.6518
FFT	0.3447	0.6107	0.6410	0.6320	x
SC	0.5425	0.6485	0.7964	0.7820	x

TABLE 2: Accuracy for Valence and Arousal labels

needed for the recurrence, a Gated Recurrent Unit (GRU) layer is applied with 1024 units. After this recurrent layer, the output is added. The GRU layer is a particular type of gating mechanism similar to the Long Short-Term memory, but optimized with respect to it. This mechanism is composed by connections that bring the output of a defined layer to be added to the input of the layer in the subsequent step. With this network is possible to understand if there is an useful correlation for the classification between subsequent elements in the dataset and, for this reason, they are usually powerful for modeling sequence data such as time series or natural language, where the knowing of the previous output can influence the subsequent analysis.

G. Residual Network

The Residual Network is a deep neural network that uses convolutional layers and particular connections, called *identity shortcut connections*, that skip one or more layers in the network. This type of connection permits to train very deep neural networks without the problem of gradient vanishing. With this model, it is also possible to evaluate if this type of addition leads to more recognizable properties for the classification. In this work, the ResNet neural network is implemented in its 34th version. This is chosen instead of the 50th one because of dimensionality problem with the smallest datasets. The building of this network followed its definition, from the input layer to the output one with four main repeated blocks,

VI. Results

In this section, results of the trained models with the features proposed are presented in order to find the best model for this classification task. In this work, since there are many models and features, for the one label case only the results for Arousal are exposed: the results obtained for Valence are pretty much the same, demonstrating that the networks have similar capabilities for these emotions. In tables 1 and 2 the accuracies obtained for Arousal and for both labels are reported for the first models: DNN, CNN and autoencoder.

A first important aspect to notice is the significant decreasing of accuracies in case of two labels with respect to the one label case. This important result can be due to the fact that the two single labels can be classified by models that analyze different aspect of the input signals. In a single network, it could be difficult to learn all the important features: this aspect does not permit to learn a single accurate model for the classification of both the labels and reduces the networks performance. Even if in absolute value the accuracies are lower, the performances obtained with the networks and the features are in the same ratio with respect to the one label case, demonstrating a certain capability of the couples feature-network to predict emotions from EEG signals.

Considering the deep neural network for all the features extracted for arousal, the best performances are reached with the signal dataset, reaching an accuracy of 94% for one label and 86% for two: this type of data leads to have separable data for the two labels and the DNN is able to find this separation. Also Principal Component Analysis permits to obtain data that are quite well separable, reaching an accuracy of 81% for one label and 66% for two. Quite high performances are also reached by Kernel PCA and statistical characteristics in one label case, with accuracies of around 71%. All the other processes lead to not accurate classifier, with accuracies between 50% and 70% in one label case and between 35% and 67% in the two label case.

The results obtained with a simple DNN can be interestingly compared with the ones obtained with the combination of autoencoder and DNN. For all the types of input fed to the networks and for both the cases analyzed, with an initial encoder for the extraction of a code of size 512, the performances increase. This result permits to demonstrate that the encoder's code generated with two subsequent convolutional layers contains much more information with respect to the one that a single fully connected layer is able to extract in the first layer of DNN. The best performances are obtained again for Signal data, with accuracy of 94% and 87%, and for KPCA.

High performances are achieved with the standard convolutional networks. In case of window and ICA, the results obtained are quite low, reaching at least 62% for one label and 58% for two labels. These bad results suggest that from these data is not possible to extract useful information with kernels of size (3, 3): bigger kernels could be tested to prove if no interesting information is at all contained or if the kernel is only too small to extract them. With CNN, the best result is obtained for Signal dataset, with accuracy of 97% is obtained for one label and 89% for two. The SC and KPCA datasets permit to have, then, an accuracy of 88% for one label and around 78% for two labels. It is important to notice that PCA, PCC and FFT perform with high accuracies, demonstrating that these statistical and frequency transformations lead again to useful information for classification.

Similar results are obtained for dilated convolutional networks, in particular with signals (accuracies of 95% and 84% respectively) but also with Statistical Characteristics dataset with accuracy of about 88% and 71%, while are quite lower for the other features, however always higher than 77% for one label. For window and ICA data, the accuracy are again very low. The absence of significant differences with respect to standard CNN make us understand that dilation does not extract more information from these data.

Different conclusions can be taken for the Locally Connected Convolutional network, since due to dimensionality issues the model has been trained only for four features. The best results, in this case, have been achieved with KPCA but in general the accuracies obtained are about 80% for one label and 70% for two. These values are in general lower with respect to the ones of CNN or CNNd: this operation, that should learn more accurate models applying different weights to each input value, is probably overfitting the dataset, without generalize the model for good predictions for new data.

Considering a general overview, so, standard convolutional neural networks achieve often the best possible results with accuracy always higher than 86% in one-label case. It results to be the best model to classify emotions when the entire signal from a channel or, with bit lower performances, when statistical analysis as PCA, KPCA, PCC, and SC or frequency domain analysis as FFT are performed.

A further comparison can be proposed between the networks derived from autoencoders and the convolutional ones. While both of them are built with convolutional operations and with the adding of a deep neural network at the end, the main difference is how the learning happens in the first phase. The autoencoder is trained with equal input and output, so to build a good general representation for the input data and not for the label. In the first phase of the convolutional learning, instead, the labels are the target of the training and what is extracted in the last layer of the convolutional block is related to both signal the associated label. In general, for all the types of dataset used, the accuracies derived with autoencoders are smaller - or at least very similar - than the best ones obtained with convolutional operations. With these extracted features, so, not all the information necessary to the classification is directly present in the data code extraction and the learning for specific label leads to best results. In general, best results are so obtained with convolutional networks.

	RNN		**ResNet**	
	Arousal	***Both***	***Arousal***	***Both***
Signal	0.8305	0.6952	0.7757	0.5728
Window	0.5376	0.4256	0.5803	0.5376
PCA	0.6379	0.3484	0.4482	0.3931
KPCA	0.6652	0.4228	0.5753	0.2443
ICA	0.5936	0.5173	0.5759	0.4536
PCC	0.7276	0.5294	0.6268	0.4438
FFT	0.7138	0.5762	0.6037	0.4761
SC	0.6206	0.5865	0.5759	0.5074

TABLE 3: Accuracy for RNN and ResNet

Another analysis can be done on the accuracies obtained with Recurrent Neural Network and Residual Networks and reported in table 3.

The performances achieved with Recurrent Neural Networks are quite good: the accuracies obtained for Arousal are still high in case of signal dataset and PCC. It reaches a value of 83% with Signal and 73% with PCC for one label, while no more than 69% is reached for two labels. Due to execution problem, it has not been possible to train this type of network with a DNN added at the end. This improvement could make the accuracy increase: it is important to underline that best accuracies obtained before the addition of a DNN in the convolutional blocks were in general higher with respect to the ones obtained in this case. As a consequence, the recurrence added by this network does not lead to the extraction of interesting hidden patterns and neither to more accurate networks. This conclusion agrees with the type of dataset presented in this study: subsequent elements feed to the network are not strictly related as a natural language phrase could be. With respect to this network, so, a convolutional network can be preferred without loss of accuracy.

Considering Residual Networks, it is possible to notice that for one label, the accuracy is often around 0.5, as a simple random classifier, and quite good classifier are reached again for Signal and PCC analysis. This two good results are not sufficient to be compared to the ones obtained in table 1. For the two labels case, this network has performances really low for all the features used. These low accuracy values are obtained for all training, validation and test sets during all the epochs of the training, demonstrating that this type of network is not able to improve its performances. The addition of the connections between layers does not permit to build a model able to perform the separation between the two or four classes. The bad results of this network could be improved with a deeper analysis, using for example a newer version.

At the end, comparing the results for the different datasets in one label case, it is possible to notice how the signal one have very high values of accuracy: this aspect can be due also by the fact that the entire dataset contains an higher number of elements with respect to the others and, in this way, the learning happens for an higher number of iterations inside each epoch. The performances achieved are however very good and the combination of convolutional and deep networks is able to well generalize classification mechanisms from the input set. From this analysis, it is possible to conclude that important information about the emotions are present in entire single signals rather than in multi-channels analysis: valence and arousal do not depend on how the different brain parts works together but more on how the signals evolves in time. The use of this dataset can be however a problem for the amount of memory necessary for the models and the high computational time. Nevertheless, using statistical processings on multi-channel dataset as statistical characteristics or Kernel Principal Components Analysis, it is possible to extract meaningful patterns for this classification task. In addition, they reduce the dataset dimensionality and, as effect, computational time and space. However, this data transformation reduces also the models accuracy, obtaining in this way pretty lower generalization on the datasets. In this analysis, KPCA is preferred with respect to PCA demonstrating that a non-linear transformation is more meaningful when multi-channel EEG signals are analyzed. The importance of SC is another key aspect: only four statistical characteristics of the data distribution generated by each channel are able to contain the pattern necessary for the classification. Similar conclusions can be taken for the two-label case: the best results are obtained with signal datasets and convolutional networks but quite high results are obtained also from the computation of statistical analysis of the data or kernel PCA, even if in all cases predicting these two labels is an harder task with respect to a single one.

VII. Concluding Remarks

In this work several features and types of neural networks have been tested to find the more appropriate for the classification of EEG signals in terms of Arousal and Valence. The analysis proposed revealed that using the entire EEG signal for the emotion prediction reveals more significant information with respect to multi-channel signals and leads to the best classifiers, reaching the highest values of accuracy in the one-label case with values up to 97%. The best models for this type of data are the combination of convolutional networks with deep fully-connected classifiers. Since this dataset is big, both in terms of number of elements and their dimension, the training and the subsequent data prediction require a lot of computational time and space. However, also the combination of data measured by different channels is able too to reveal patterns for this task, leading to good classifiers. As a matter of fact, with Statistical Characteristics of the channel and Kernel Principal Component Analysis from time-windowed data, very high performances are reached by the network. A consequence of these results is also the importance of the statistical analysis of EEG signals in the emotion classification task: these analysis permit to have an important dimensionality reduction, maintaining very high accuracies. This reduction permits a very high reduction of the computational time needed for training of the models and also for the subsequent prediction of new data.

In case of two-label task, all the accuracies obtained are smaller, revealing the difficulty of a neural network to predict both of them at the same time, and by consequence the possible different structure that is above behind the two different emotions. The results obtained for them are quite similar to the one-label cases: signals have best performances with accuracy of 89% but very high computational time and space needed; other statistics as SC or KPCA perform quite well, reducing the computation needed.

The analysis proposed in this paper can be useful for future works, since, looking for more accurate models for this task, the construction of a neural network can start from the ones reported above with the features that lead to best results, or combination of them. In fact, combining these features or model could reveal more hidden data patterns. An example could be the training of an encoder, followed by a convolutional neural network and a final deep network, each trained as described above: the first part could extract the most important characteristic of input data, the second could determine hidden features of them thanks to the kernels applications and the last could be an accurate classifier. The best models could be also trained for an highest number of epochs, to arrive to a more precise set of network weights. Another future works could be the implementation of the deep neural network after the recurrent neural network or the analysis of the ICA transformation applied to the entire signal rather than on time windows seeing the poor performances reached with this dataset.

This work is important when a signal has to be analyzed with a neural network: select the proper feature and type of network is a necessary step in order to build a good model. A wrong selection could in fact lead to very low performances, as happened with ResNet, even if the data are classifiable in a good way. A difficulty that can be encountered in this procedure is, however, the very high computational time and resources needed for each training.

References

[1] S. Koelstra, C. Mühl, M. Soleymani, J. S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "DEAP: A Database for Emotion Analysis using Physiological Signals," *IEEE Transactions of affective computing*, vol. 3, pp. 18–31, June 2012.

[2] J. X. Chen, P. W. Zhang, Z. J. MAO, Y. F. Huang, D. M. Jiang, and Y. N. Zhang, "Accurate EEG-Based Emotion Recognition on Combined Features Using Deep Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 44317–44328, Apr. 2019.

[3] J. Liu, G. Wu, Y. Luo, S. Qiu, S. Yang, W. Li, and Y. Bi, "EEG-Based Emotion Classification Using a Deep Neural Network and Sparse Autoencoder," *Frontiers in Systems Neuroscience*, vol. 14, no. 43, pp. 1–14, 2020.

[4] S. Tripathi, S. Acharya, R. D. Sharma, S. Mittal, and S. Bhattacharya, "Using Deep and Convolutional Neural Networks for Accurate Emotion Classification on DEAP Dataset," in *AAAI'17: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence Pages*, pp. 4746–4752, Feb. 2017.

[5] A. Alhagry and A. A. Fahmy and R. A. El-Khoribi, "Emotion Recognition based on EEG using LSTM Recurrent Neural Network," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 10, pp. 355–358, 2010.