J Seismol
https://doi.org/10.1007/s10950-021-10055-8

ORIGINAL ARTICLE

Automatic inspection and analysis of digital waveform images by means of convolutional neural networks

Alessandro Pignatelli ·
Francesca D'Ajello Caracciolo · Rodolfo Console

Received: 31 May 2021 / Accepted: 5 October 2021

Abstract Analyzing seismic data to get information about earthquakes has always been a major task for seismologists and, more in general, for geophysicists. Recently, thanks to the technological development of observation systems, more and more data are available to perform such tasks. However, this data "grow up" makes "human possibility" of data processing more complex in terms of required efforts and time demanding. That is why new technological approaches such as artificial intelligence are becoming very popular and more and more exploited. In this paper, we explore the possibility of interpreting seismic waveform segments by means of pre-trained deep learning. More specifically, we apply convolutional networks to seismological waveforms recorded at local or regional distances without any pre-elaboration or filtering. We show that such an approach can be very successful in determining if an earthquake is "included" in the seismic wave image and in estimating the distance between the earthquake epicenter and the recording station.

Keywords Machine learning · Image processing · Neural networks · Computational seismology · Instrumental noise

A. Pignatelli (✉) · F. D'Ajello Caracciolo · R. Console
Istituto Nazionale Di Geofisica E Vulcanologia (INGV), Rome, Italy
e-mail: alessandro.pignatelli@ingv.it

1 Introduction

In the recent decades, the continuous improvement of quality and quantity of seismographic equipment has increased the amount of data produced by seismological networks in such a way that its visual analysis by human analysts has become an impracticable task. At the same time, the progress in computer technology has allowed the development of automatic processing tools, among which deep neural networks have shown the capacity of supporting in an efficient way the human capacities.

The application of neural networks to waveform analysis to detect seismic signals from the background noise, phase picking, and event location has started in the 1990s with the early work of Romeo (1994) and Wang and Teng (1995, 1997). Following these pioneristic efforts, a quantity of papers has populated the scientific literature (Gentili and Michelini 2006; Tiira 1999; Zhao and Takano 1999). In particular, significant results have been achieved in the most recent studies by Perol et al. 2018; Li et al. 2018; Chen et al. 2019; Kriegerowski et al. 2019; Lomax et al. 2019; Mosher and Audet 2020.

In this paper, we propose an innovative approach using just seismic waveform images rather than time sequences as input to neural networks. More specifically, the waveform analysis consists in two steps as follows:

- Training a convolutional neural network to automatically classify digital waveform segments col-

Published online: 14 October 2021

lected by the Italian Seismological Instrumental and Parametric Database (ISIDe) of the Istituto Nazionale di Geofisica (INGV) (ISIDe Working Group 2007)

- Training a regression neural network to estimate the distance between the earthquake epicenter and the station recording the signal, from waveform segments previously classified by the previous step

2 Methodology used

Neural networks are a subset of machine learning techniques. By definition, "machine learning" is meant to be the use and the development of computer systems able to achieve specific tasks by means of a learning process rather than a set of explicit instructions. More specifically, artificial neural networks (ANN) are algorithms inspired by the structure of the primate cerebral cortex designed to learn abstract features from the input data to support the desired output (O'Shea and Nash 2015; Rawat and Wang 2017).

The most general problem solvable by a neural network is trying to predict a variable (said *answer* or *label*) when some other variables (said *predictors*) are known. For example, in a seismological context, the predictors could be a single waveform segment and the label could be the classification if the total signal contains an earthquake or not. In this example, the main task of a neural network would be to "learn" if provided data contains proper features to perform a correct classification.

Any "learning process" requires a *training* step and specific data to learn from. That is why the first effort when dealing with neural networks is providing the system with data containing both the predictors and the labels. Once such data are provided and the system has been successfully trained, then the algorithm is able to automatically predict the answer for any other data sets including the same kind of predictors.

Depending on the type of answer, neural networks can be divided into two groups. If the output variable is categorical or included in a limited set of discrete values, then the network is said to be a *classification* one. Basically, a classification network is characterized by groups of possible outputs and it has to learn how to determine which group is the right one for the input data. In our example, the groups are "earthquake" and "not earthquake." So, the specific example would require a classification network. If the answer is a continuous number, then the network is said to be a *regression* one. The regression networks are designed to learn a mathematical dependency between the input and output variable so that the output number can be calculated according to the input data.

A common practice to evaluate the "prediction capability," once the training process is finished, is using "test data." Basically, a randomly chosen fraction of input data and their corresponding labels provided by the user are excluded from the training process, so that the "already trained" system can be applied to them getting their "predicted labels." For this data, both "predicted" and "real" labels are available and can be compared to estimate the accuracy of the model. An additional practice used to better check "prediction capability" is the *k*-fold procedure. Basically, rather than simply dividing the available labelled data into two groups (training and test), a set of test data groups are selected and multiple training processes are performed, using each group as test data and the remaining data as training data. So, for example, if 20% of percentage is established for test data, then a fivefold procedure is applied: all data are divided into five groups (each of them having 20% of data) and five training processes are used using the remaining 80% of data for training and the 20% of that group as test. This allows a better estimate of method accuracy.

A general schema of ANN is shown in Fig. 1 as described in Romeo (1994). The basic ANN elements are neurons inserted into proper layers and connected to all or to a partial subset of other neurons of the network. The first layer is defined "input layer" and its neurons' number is the same as input data so that each single input is "passed" to a single neuron and then, after convenient mathematical steps, passed to the next layer where its processing goes on till the last layer, where the output can be compared to the provided labels to check their match.

All mathematical steps involve ANN parameters called *weights* and *biases*, and the final output is strongly dependent on their values. When the training process starts, such values are randomly chosen and, after comparing the predicted outputs with the provided labels, a *backward* process starts modifying the weights and the biases. Such propagation involves the iterative adjustment of the parameters vector with the goal of minimizing the differences between the observed and predicted values (Cao and Parry 2009).

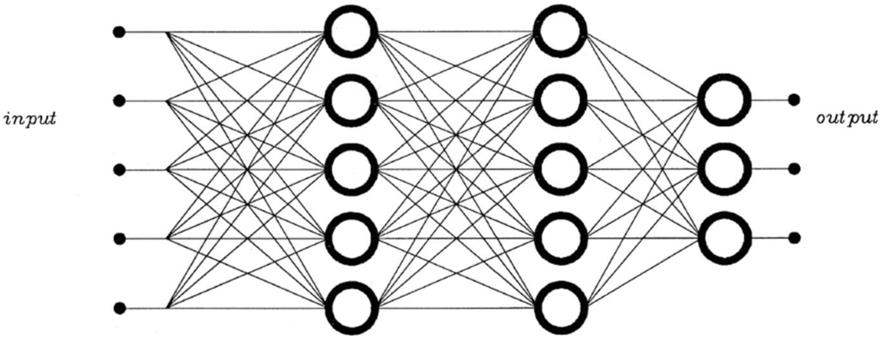


Fig. 1 General neural network architecture (Romeo 1994)

Convolutional neural networks (CNNs) are a subset of ANN specifically designed to extract local features from matrices. Their structures are more complex and their layers basically consist of filters applied to matrices (Cao and Parry 2009; Indolia et al. 2018). The main task of CNN is to recognize "local" rather than "global" features. This means that each neuron, rather than collecting data coming from all inputs (or neurons of the previous layer), focuses just on a limited number of them. To accomplish this task, CNN consists of sequences of four layer types: convolutional layers, pooling layers, fully connected layers, and a single softmax layer.

Convolutional layers are the most important to inspect local features and their characteristics can be summarized in Fig. 2.

As one can see, just a limited number of inputs/neurons are connected to each neuron of the next layer. By means of such strategy, each neuron can inspect specific areas. The convolutional layer outputs are matrices.

Pooling layers are designed to subsample the convolutional layers output in order to reduce their sizes. An example is shown in Fig. 3 showing a polling layer output. The final matrix contains the maximum value of all 2×2 submatrices starting from matrix layer input. Rather than maximum, it is possible to use different functions such as minimum, average, and others.

Fully connected layers are basically the ones shown in Fig. 1 and they are inserted into the CNN to change the output matrices coming from convolutional and pooling layers from a 2d structure to a 1d one.

Generally, the first fully connected layer in a CNN is followed by other fully connected layers as shown in Fig. 4.

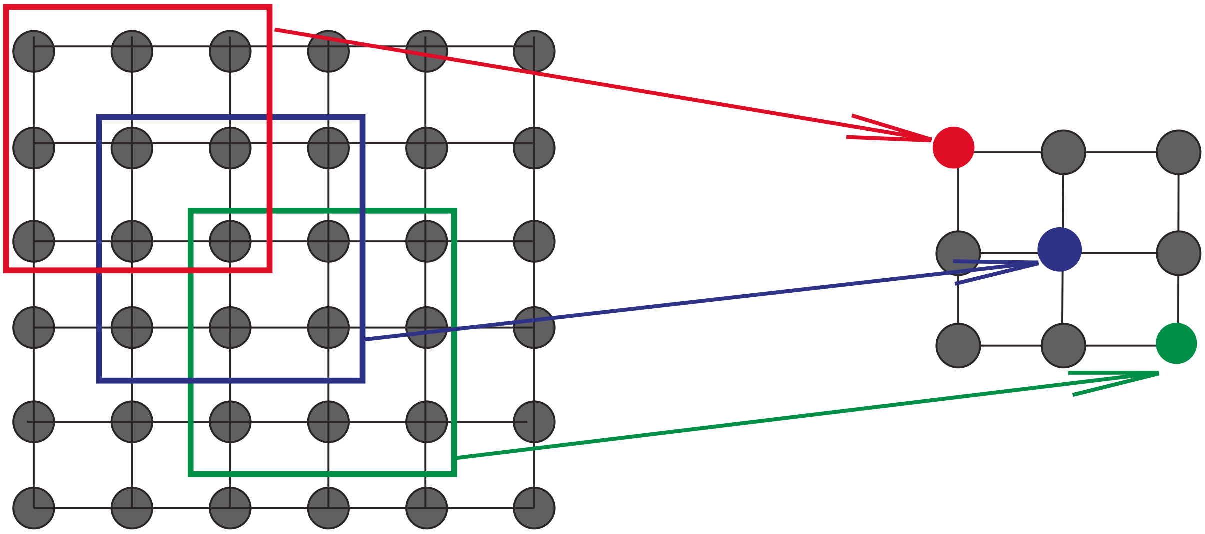
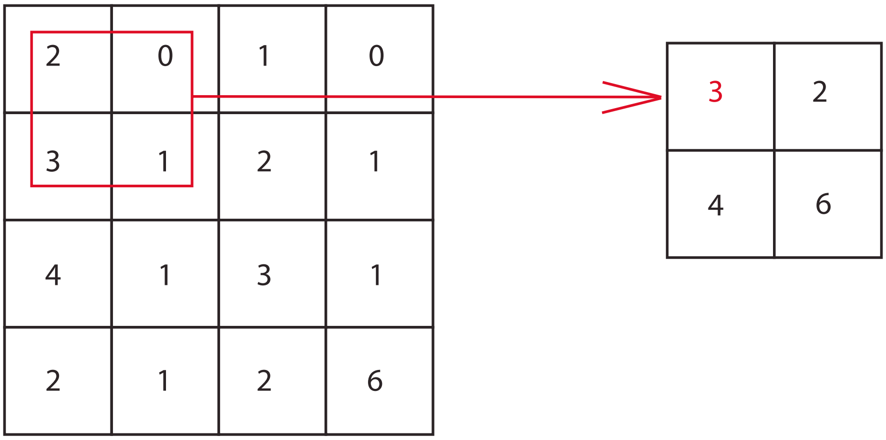


Fig. 2 Convolutional layer representation

Fig. 3 Pooling layer representation



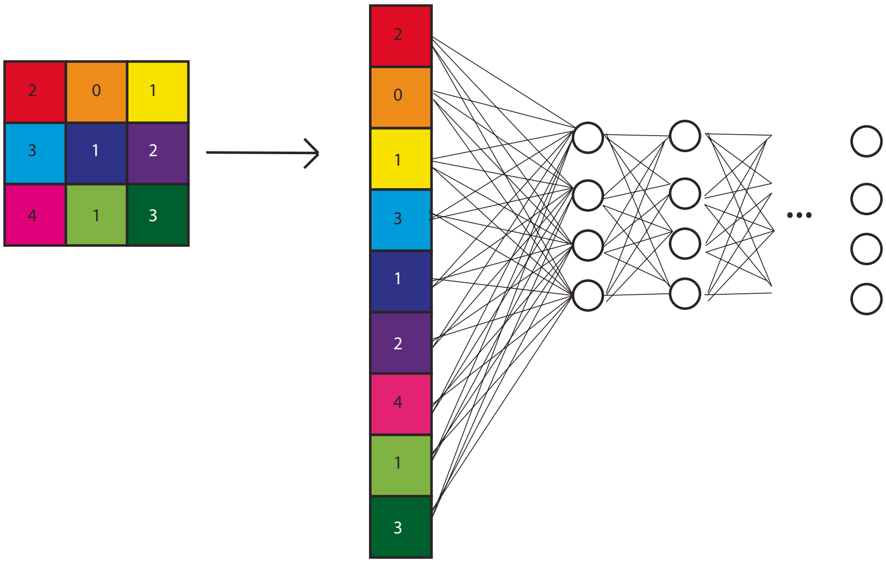
Softmax layers transform the numbers included into the last fully connected layer into probabilities of each label. Finally, the predicted label is the one correspondent to the higher softmax value.

Convolutional networks have extensively been used to classify images and a very famous one is *Alexnet* (Han et al. 2017; Indolia et al. 2018; Krizhevsky et al. 2017). The main capability of Alexnet is extracting meaningful features from images. Such features allow the final layers of the network to be used for different classifications or regression problems. The main advantage of using Alexnet is that this is a pre-trained network and the weights are already pre-calculated to recognize primitive image features (such as geometrical figures, segments, and so on). So, the network is, for a large part, already trained and optimized to be used and the training step consists just in tuning the pre-calculated parameters. This ensures a quick and optimal training compared to a general network designed from scratch. Furthermore, a smaller number of input data have to be provided compared to a new neural network.

Finally, using simple and few proper modifications, Alexnet can be used for both purposes: classification and regression. More specifically, replacing the softmax layer with a single fully connected layer, it is possible to transform the classification Alexnet architecture into a regression one; both architectures are shown in Fig. 5.

Another important concept to introduce when describing neural networks is the “loss function.”

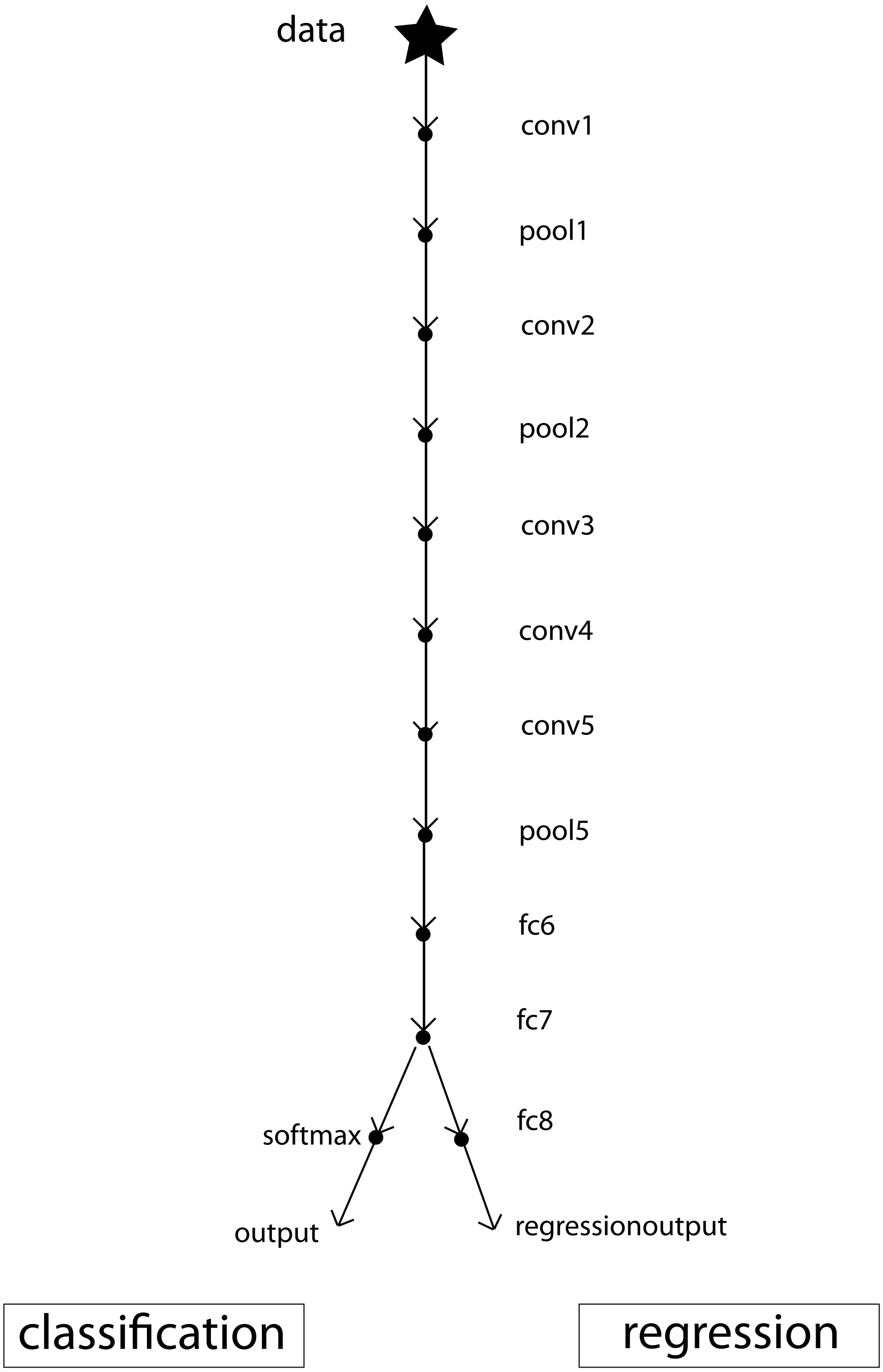
Fig. 4 Fully connected layers representation



Basically, the loss function determines, at each iteration of training progress, how much the labels predicted by the network are different from the real ones (they are known for training data). Such "loss functions" may also have complicated expressions but, for Alexnet, the functions used to define the loss are cross-entropy (Ho and Wookey 2019) for the classification and the mean-squared error for regression.

There are additional pre-trained networks (such as VGG and Resnet) that could be used, but in this paper, we decided to use Alexnet as it is less computing time and hardware resources demanding. However, we have

Fig. 5 Alexnet architecture representation for both classification and regression



also checked that Resnet and VGG have similar or worse results than Alexnet (see additional material at https://gitlab.com/alessandro.pignatelli/automaticinspectionmaterial/-/blob/main/auxiliary_material.zip).

In this work, we show that three components seismic wave plots can be used (just as images without using digital waveforms) to feed an Alexnet neural network to be trained in order to perform the following two tasks:

1) Determine if a waveform segment contains an earthquake signal (and more specifically if the source of such signal is near of far from the recording station).
2) Estimate the distance between the seismic event and the stations recording the waveform.

We show that results are accurate enough to provide a general neural network for an automatic detection system.

3 Data and resources

We used 8348 three-component waveforms from earthquakes with magnitude > 3 that occurred in the Italian peninsula from 2015 to the present and recorded at seismic stations of the INGV network. The seismic network of INGV consists of broadband, short period, and high gain seismometers. The waveforms were automatically downloaded from the Italian Seismological Instrumental and Parametric Database (ISIDe, http://terremoti.ingv.it/iside) by means of web scraping procedures. As stated in the previous section, in this work, we have accomplished two tasks. For both of them, our “input data” have been plotted as the one shown in Fig. 6: basically, the three components of a seismic wave recorded by INGV stations for a period of 5 min. We used three-component seismograms as it is well known that the vertical component is the most useful to detect the P wave first motion and the horizontal components show more clearly S waves arrivals. It could have some relevance in the task of estimating the epicentral distance from the full set of three components.

The image has been intentionally left without any title or label to avoid network being influenced by parametric data during the training process.

Such images have been used for both work tasks. More specifically, for the first task, a set of 2700 images have been manually selected and divided into three groups: “yes local” group (signals showing a

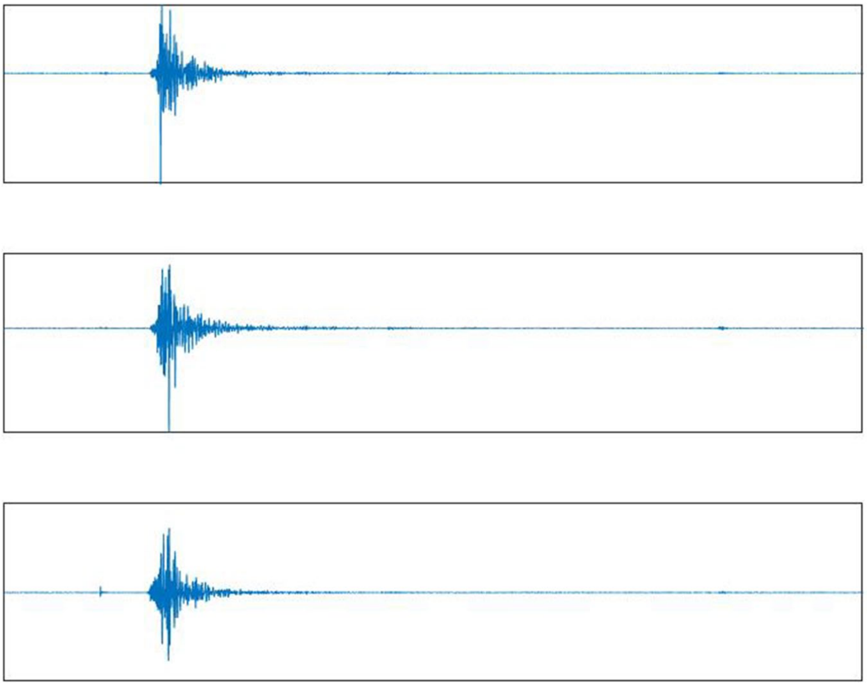


Fig. 6 An example of a neural network input image. It contains the three components (E-W, N-S, and vertical) of seismic waves. The image has been intentionally left without any axes labels to avoid numbers influencing the neural network training process

Fig. 7 Accuracy and confusion matrix obtained by the Alexnet classification network for the first fivefold step

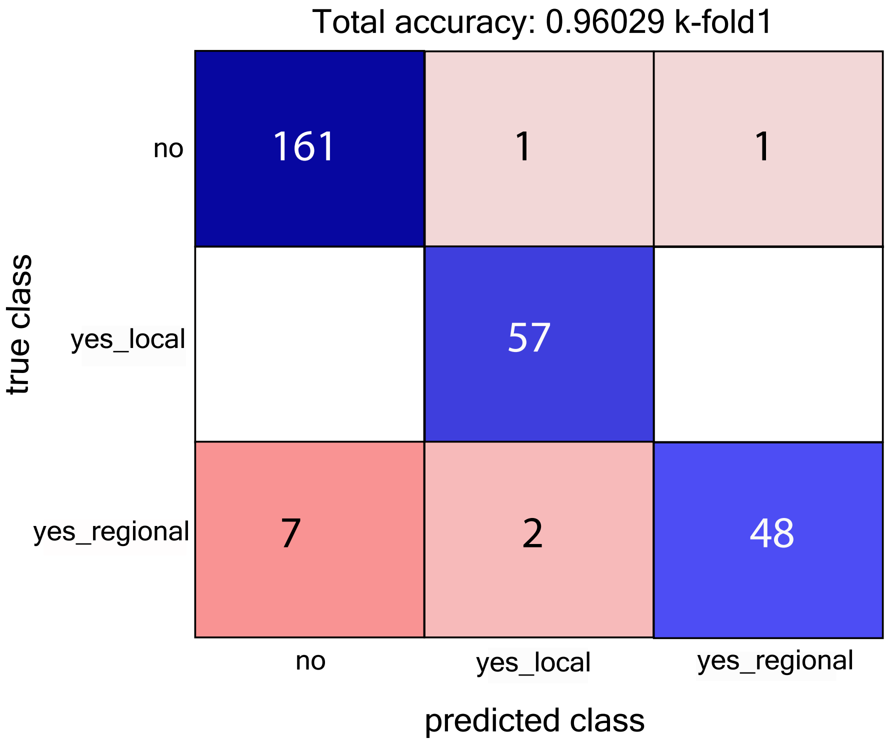
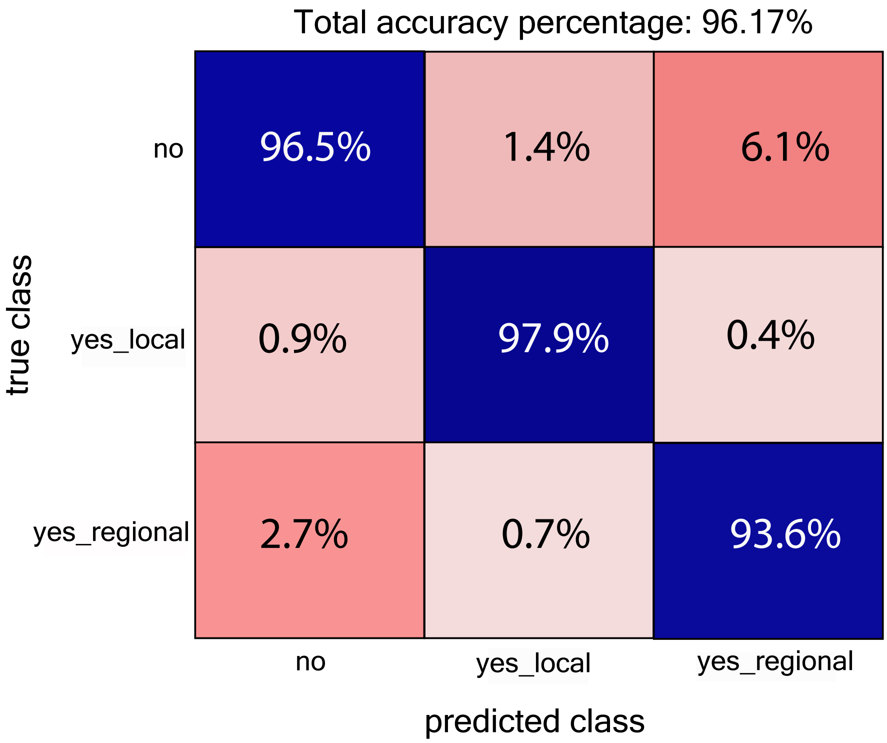


Fig. 8 Accuracy and confusion matrix obtained by the Alexnet classification network and averaged on all the fivefold procedure steps



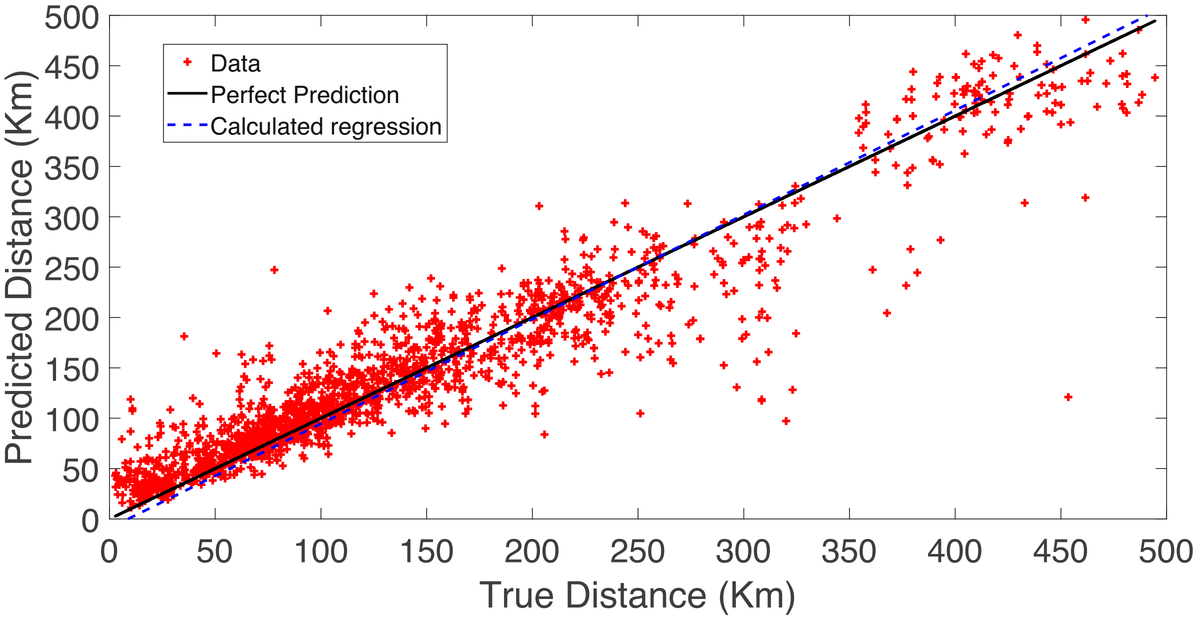


Fig. 9 Neural network regression results for the first fivefold step. The black line shows the "ideal" perfect regression. The red spots show the true values vs the network predicted ones. The dashed line is the result of linear regression between predicted distance values and the observed ones obtained from the ISIDe database

clear earthquake at less than 150 km from the recording stations), "yes regional" group (signals showing a clear earthquake at more than 250 km from the recording station), and "no" group (signals not showing a clear earthquake or too much disturbed by noise). More specifically, in order to balance the "event" and "no event" samples, we have chosen 1626 "no," 571 "yes local," and 573 "yes regional."

Such images have been used for the first step of this work. So, the Alexnet network has been trained to learn how to perform the image group assignment.

Once this process was completed, we used the trained network to automatically select three-component seismograms to perform the second processing task for a total of 8348 samples.

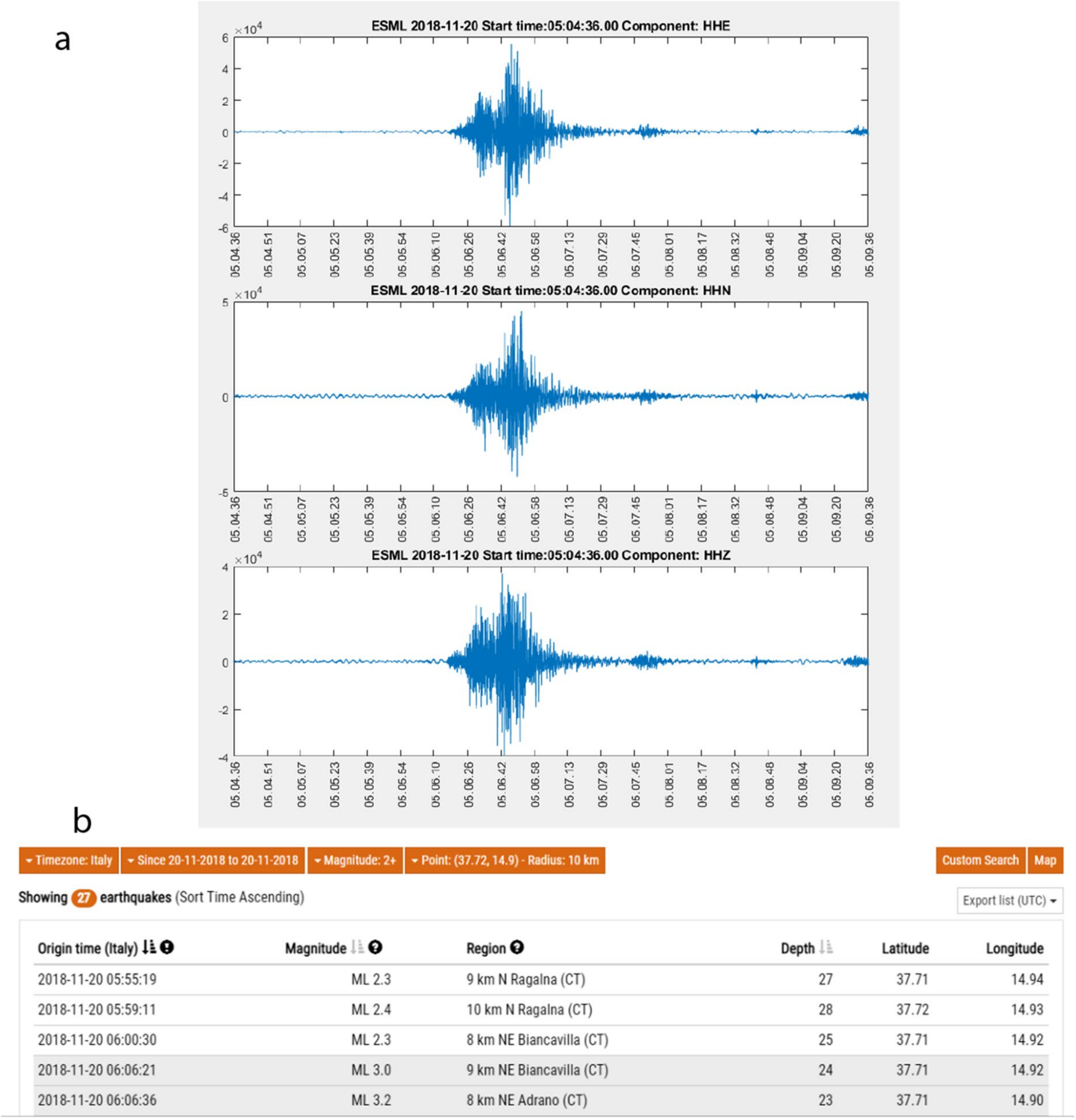
More specifically, some earthquakes have been randomly chosen from the ISIDe database and (starting from the nearest to the farthest station of the Italian network) three seismograms of 5 min length have been downloaded from each station. The starting record points have been randomly selected in a range of 5 min before the origin time of the respective earthquake. This "random shift" has been inserted to avoid that the network could be influenced by the distance from the starting recording point and the first arrival of the seismic waves as, in a possible automatic process, the earthquake origin time is unknown and the time windows passed to the network would start from a random point.

For each downloaded data set, we have also recorded the distance from the earthquake epicenter and the station (so we could have both input data and their labels), and for the purpose of this work, we have excluded the signals recorded by the stations more distant than 500 km.

Once the automatic download process has been completed, we have first filtered data by means of the first step trained network (so only earthquakes classified data have been selected) and then used the downloaded signals to feed the Alexnet network and trained it to estimate the distance from the image.

4 Results

For both tasks described in the previous section, we decided to use 20% of the available data from the test process. To better estimate method accuracies, we also used a fivefold procedure, so we have performed 5 times the training process and each time, we used



▾ Timezone: Italy ▾ Since 20-11-2018 to 20-11-2018 ▾ Magnitude: 2+ ▾ Point: (37.72, 14.9) - Radius: 10 km

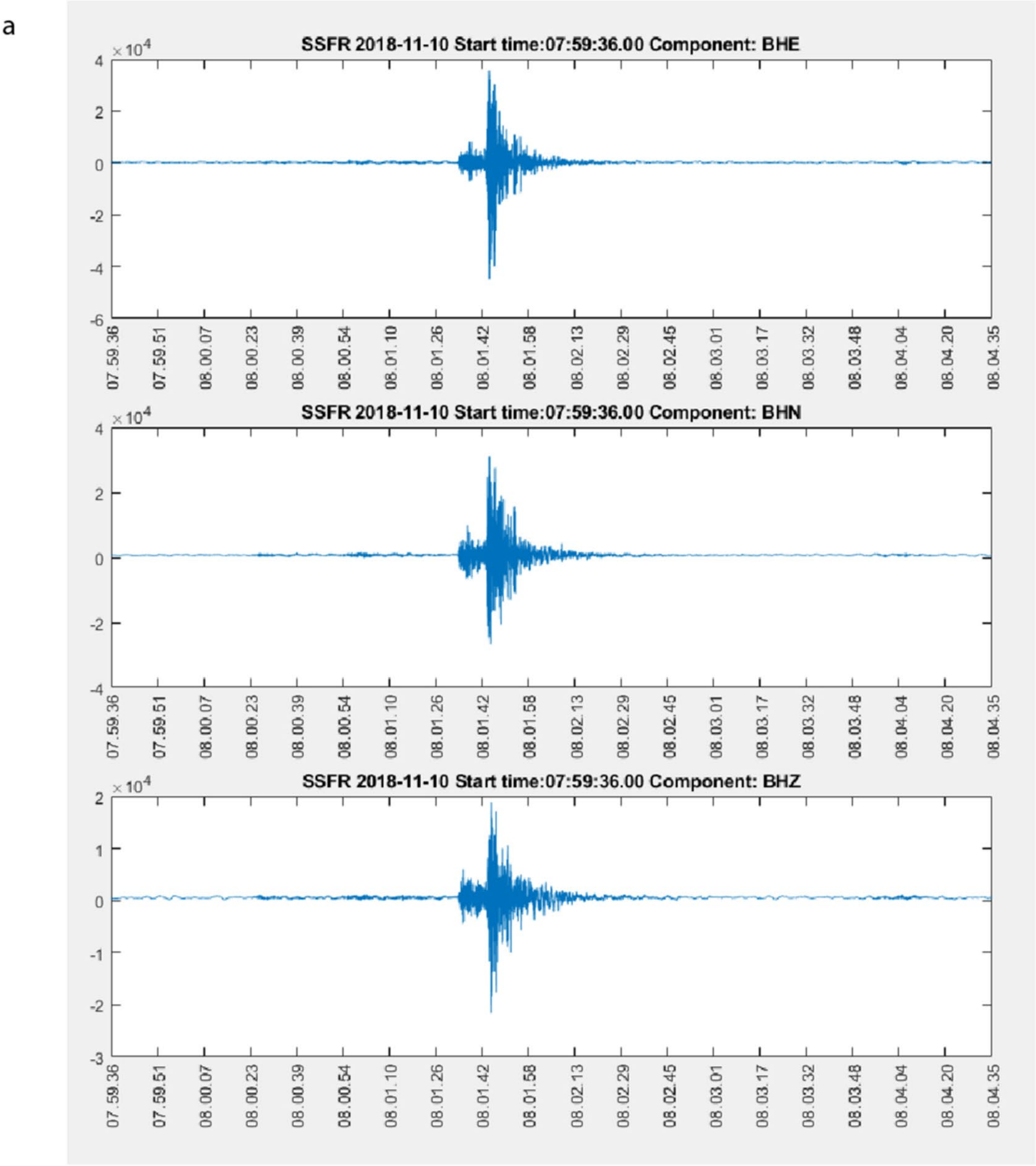
Showing 27 earthquakes (Sort Time Ascending)

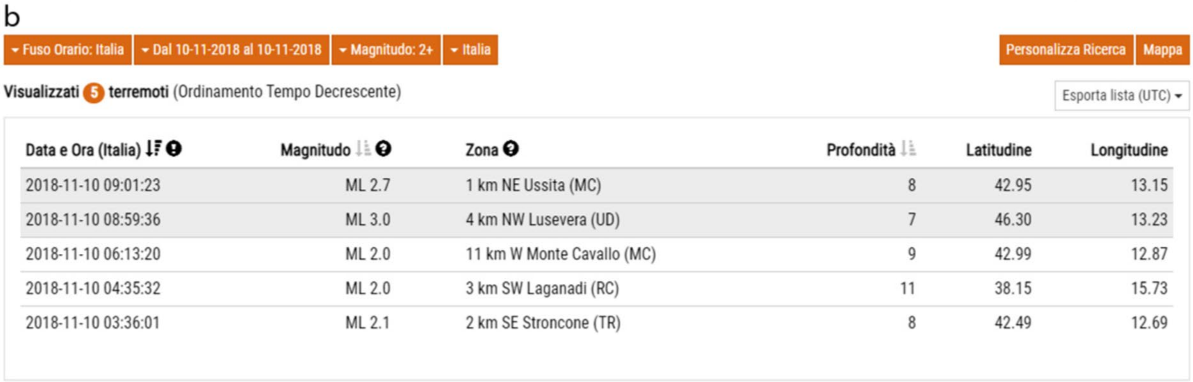
Origin time (Italy)	Magnitude	Region	Depth	Latitude	Longitude
2018-11-20 05:55:19	ML 2.3	9 km N Ragalna (CT)	27	37.71	14.94
2018-11-20 05:59:11	ML 2.4	10 km N Ragalna (CT)	28	37.72	14.93
2018-11-20 06:00:30	ML 2.3	8 km NE Biancavilla (CT)	25	37.71	14.92
2018-11-20 06:06:21	ML 3.0	9 km NE Biancavilla (CT)	24	37.71	14.92
2018-11-20 06:06:36	ML 3.2	8 km NE Adrano (CT)	23	37.71	14.90

Fig. 10 **a** Two local earthquakes of magnitudes 3.0 and 3.2, origin time 15 s apart, recorded by station ESML (Etna Volcano). True distance: 10 km; predicted distance: 103 km; **b** the INGV (ISIDe) seismic bulletin reporting the two events (highlighted in gray) recorded in Fig. 10a

20% data for testing to estimate the capability of the two trained networks to generalize the prediction to an independent data set.

Common ways to estimate classification networks prediction are the accuracy (defined as the ratio between the number of correctly classified data and the total data) and the confusion matrix (Lantz 2013). A confusion matrix is a table counting when the predicted classes agree or disagree with the true values. Conventionally, the rows represent the true classes and the columns represent the predicted ones. The number in the intersection between rows and columns





Visualizzati 5 terremoti (Ordinamento Tempo Decrescente)

Data e Ora (Italia)	Magnitudo	Zona	Profondità	Latitudine	Longitudine
2018-11-10 09:01:23	ML 2.7	1 km NE Ussita (MC)	8	42.95	13.15
2018-11-10 08:59:36	ML 3.0	4 km NW Lusevera (UD)	7	46.30	13.23
2018-11-10 06:13:20	ML 2.0	11 km W Monte Cavallo (MC)	9	42.99	12.87
2018-11-10 04:35:32	ML 2.0	3 km SW Laganadi (RC)	11	38.15	15.73
2018-11-10 03:36:01	ML 2.1	2 km SE Stroncone (TR)	8	42.49	12.69

◂**Fig. 11** **a** Local event of magnitude 2.7 recorded at 61 km from the SSFR station masked the recording of a farther event of magnitude 3.0 at a distance of 320 km; **b** the INGV (ISIDe) seismic bulletin reporting the events (highlighted in gray) recorded in Fig. 11a

represent how many records characterized by the row true classes have been classified with the predicted one specified by the column. So the numbers along the diagonal show all the records correctly classified and the other numbers the wrong cases. In addition to the overall accuracy, confusion matrices indicate the accuracy of the method for each single class. For our experiment, both accuracy and confusion matrix are shown in Fig. 7 (the first matrix obtained by the fivefold procedure). All other k-fold results are very similar, as one can see at https://gitlab.com/alessandro.pignatelli/automaticinspectionmaterial/-/blob/main/auxiliary_material.zip. In our case the accuracy is 96.03%, showing that using images rather than time sequences does not affect at all the neural network prediction capability. To resume all the results of the fivefold procedure, we computed the average percentage confusion matrix shown in Fig. 8.

For the second task, the results are shown by the plot of Fig. 9 (the first regression obtained by the fivefold procedure. All other k-fold results are very similar, as one can see at https://gitlab.com/alessandro.pignatelli/automaticinspectionmaterial/-/blob/main/auxiliary_material.zip), where the black line shows the ideal "perfect prediction line" and the red spots show the actual predictions. As one can see, there is a good match between real and predicted distances (especially for the lowest ones which are the most important for a possible automatic system) and we have checked that for each very wrong estimate, there is a specific reason explaining such error as shown in Sect. 5.

In order to estimate the regression quality, we performed a linear regression using the real distances as independent variable and the predicted distances as dependent variable. If the neural network worked perfectly, we would expect a regression line passing through the origin and with angular coefficient 1 (black line of Fig. 9). As we can see from Fig. 9 (dashed line), the real regression line is very near to the ideal "perfect prediction line." In fact, the calculated coefficients are linear term = 1.0371 and intercept = −9.1424 km. Moreover, the *t* Student statistical test applied to the coefficients gives *p* values very near to zero (more specifically both for the linear term and the intercept the values are lower than 1e-27) meaning that the linear relation is statistically highly significant even if the scatter plot shows some outliers." Furthermore, we estimated the data scattering level by computing the percentage error median 13.89% and the interquartile range 22.88%.

5 Discussion and conclusions

In order to understand and classify the network errors, we have analyzed some of the outliers, i.e., the cases in which the value of the epicentral distance predicted by the neural network was largely different from the real value reported on the ISIDe bulletin. As an example, in Figs. 10 and 11, we report a couple of anomalous cases; in one case (Fig. 10a), two earthquakes of respective magnitudes 3.0 and 3.2 occurred 15 s apart in the same epicentral area, so that the neural network has instead regarded them as a single event considering the time interval between the two first onsets as the difference between the arrival of P waves and that of S waves. This happened in the area of the Etna volcanic district (as reported by the INGV bulletin in Fig. 10b) where seismic sequences are frequent (Alparone et al. 2015).

Another common case (Fig. 6a), where the neural network has proven some difficulties in providing the expected distance, is when a small but nearby earthquake is also recorded at a station, masking the trace of an almost simultaneous but much farther event, even if the latter has a larger magnitude. In the specific case shown in Fig. 11, the SSFR station received a seismic signal from a nearby event (at 61 km distance) of magnitude 2.7, which occurred 2 min after an earthquake of magnitude 3.0 located 320 km far from the same station. The latter event was the one to be identified by the automatic testing procedure. In this case, the predicted distance calculated by our neural network was 74 km. The presence of the magnitude 2.7 earthquake was confirmed at other nearby stations, so that the distance provided by the neural network cannot be considered a real mistake.

In conclusion, in this study, we have tested the hypothesis that data images of three-component seismic waveforms contain much of the information included in time series or, at least, the information required to classify if a seismic event is included or not in the data set under analysis and to give an approximate estimate of the distance between the seismic event epicenter and the recording station. To prove such a claim, we applied a deep neural network algorithm to plots of waveform segments obtained from the ISIDe database of the Italian INGV seismological network to classify the presence of seismic events inside a waveform segment and to estimate the respective epicentral distance. The results show a classification accuracy of about 96% and a very good fit in terms of distance prediction capability.

We performed the same analysis also using other pre-trained networks such as VGG or Resnet. We got very similar results for regression while, for classification VGG performance, it seems to be lower than the other two. We put such additional results together with Alexnet one into auxiliary material at https://gitlab.com/alessandro.pignatelli/automaticinspectionmaterial/-/blob/main/auxiliary_material.zip.

Acknowledgements We are very thankful to the anonymous referee for his/her valuable comments and precious suggestions to improve our research. We are also grateful to the Editor in Chief Mariano Garcia-Fernandez for handling our manuscript.

Author contribution All the authors have contributed to all the paper content parts and its drafting. Moreover, the manuscript is the result of continuous interaction, proposal, and discussion among them. More specifically, Alessandro Pignatelli has given most of his contribution in designing the neural network architecture and in coding the procedures to run them with data; Francesca D'Ajello Caracciolo has contributed mostly in collecting data, analyzing, and classifying the input waveform to feed the classification network and to refine plots; and Rodolfo Console has mostly taken care of seismological issues and overall revision of the manuscript.

Funding Open access funding provided by Istituto Nazionale di Geofisica e Vulcanologia within the CRUI-CARE Agreement.

References

Alparone S, Maiolino V, Mostaccio A, Scaltrito A, Ursino A, Barberi G, D'Amico S, Di Grazia G, Giampiccolo E, Musumeci C (2015) Instrumental seismic catalogue of Mt. Etna earthquakes (Sicily, Italy): ten years (2000–2010) of instrumental recordings. Ann. Geophys 58:0435

Cao Q, Parry ME (2009) Neural network earnings per share forecasting models: a comparison of backward propagation and the genetic algorithm. Decis Support Syst 47:32–41

Chen Y, Zhang G, Bai M, Zu S, Guan Z, Zhang M (2019) Automatic waveform classification and arrival picking based on convolutional neural network. Earth Space Sci 6:1244–1261

Gentili S, Michelini A (2006) Automatic picking of P and S phases using a neural tree. J Seismol 10:39–63

Han X, Zhong Y, Cao L, Zhang L (2017) Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. Remote Sens 9:848

Ho Y, Wookey S (2019) The real-world-weight cross-entropy loss function: modeling the costs of mislabeling. IEEE Access 8:4806–4813

Indolia S, Goswami AK, Mishra SP, Asopa P (2018) Conceptual understanding of convolutional neural network-a deep learning approach. Procedia Comput Sci 132:679–688

ISIDe Working Group (2007) Italian Seismological Instrumental and Parametric Database (ISIDe). Istituto Nazionale di Geofisica e Vulcanologia (INGV). https://doi.org/10.13127/ISIDE

Kriegerowski M, Petersen GM, Vasyura-Bathke H, Ohrnberger M (2019) A deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms. Seismol Res Lett 90:510–516

Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60:84–90

Lantz B (2013) Machine learning with R. Packt publishing ltd

Li Z, Meier M-A, Hauksson E, Zhan Z, Andrews J (2018) Machine learning seismic wave discrimination: application to earthquake early warning. Geophys Res Lett 45:4773–4779

Lomax A, Michelini A, Jozinović D (2019) An investigation of rapid earthquake characterization using single-station waveforms and a convolutional neural network. Seismol Res Lett 90:517–529

Mosher SG, Audet P (2020) Automatic Detection and location of seismic events from time-delay projection mapping and neural network classification. J Geophys Res Solid Earth 125, e2020JB019426

O'Shea K, Nash R (2015) An introduction to convolutional neural networks. ArXiv Prepr. ArXiv151108458

Perol T, Gharbi M, Denolle M (2018) Convolutional neural network for earthquake detection and location. Sci Adv 4:e1700578

Rawat W, Wang Z (2017) Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput 29:2352–2449

Romeo G (1994) Seismic signals detection and classification using artificial neural networks. Ann Geophys 37

Tiira T (1999) Detecting teleseismic events using artificial neural networks. Comput Geosci 25:929–938

Wang J, Teng T (1997) Identification and picking of S phase using an artificial neural network. Bull Seismol Soc Am 87:1140–1149

Wang J, Teng T-L (1995) Artificial neural network-based seismic detector. Bull Seismol Soc Am 85:308–319

Zhao Y, Takano K (1999) An artificial neural network approach for broadband seismic phase picking. Bull Seismol Soc Am 89:670–680

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.