

JGR Solid Earth

RESEARCH ARTICLE

10.1029/2020JB021566

Special Section:

Machine learning for Solid Earth observation, modeling and understanding

Key Points:

- Two-stage pipeline consisting of 1D convolutional neural networks for the earthquake detection, localization, and characterization from single-station three-component waveforms
- Understanding what makes a difference when training a CNN for earthquake detection and characterization in terms of different CNN hyperparameters, classes, data sets
- Successfully detected 273 new local events using our two-stage CNN pipeline for 1-month test period

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

J. Majstorović,
josipa.majstorovic@univ-grenoble-alpes.fr

Citation:

Majstorović, J., Giffard-Roisin, S., & Poli, P. (2021). Designing convolutional neural network pipeline for near-fault earthquake catalog extension using single-station waveforms. *Journal of Geophysical Research: Solid Earth*, 126, e2020JB021566. <https://doi.org/10.1029/2020JB021566>

Received 16 DEC 2020

Accepted 24 JUN 2021

Designing Convolutional Neural Network Pipeline for Near-Fault Earthquake Catalog Extension Using Single-Station Waveforms

Josipa Majstorović¹ , Sophie Giffard-Roisin¹ , and Piero Poli¹ 

¹Institut des Sciences de la Terre, Université Grenoble Alpes, CNRS (UMR5275), Grenoble, France

Abstract In this study, we developed an end-to-end two-stage pipeline using 1D convolutional neural networks (CNNs) to detect, localize, and characterize earthquakes from single-station three-component waveforms. We are presenting here the insights of what makes the difference in developing a deep learning algorithm by performing an extensive hyperparameter grid search for model training, tackling the question of the optimal number of classes, the importance of training data sets as well as the CNN architecture design in terms of optimal length of the CNN model. Moreover, our pipeline is robust and does not need any preprocessing of the seismograms (e.g., filtering) or any prior knowledge of the region. Training, validation, and evaluation of the CNN models is performed on data recorded at the AQU station placed in the city of L'Aquila in the Abruzzo region (Central Italy). Before MW 6.3 2009 L'Aquila earthquake that occurred near the city of L'Aquila the local catalog of events is sparse. Therefore, we provide a methodological pipeline on how one can extend near-fault local catalog of earthquakes by applying our two-stage pipeline on unseen continuous data. Our results show that we are able to design a CNN model that is detecting the earthquake events among random noise waveforms with 97% accuracy (first stage: detection). Additionally, we are able to determine the events that are close to the station (<10 km) with a 94% accuracy as well as identify their belonging to four magnitude classes with a 68% accuracy (second stage: characterization).

Plain Language Summary In this study, we developed a deep learning pipeline for earthquake detection, localization, and characterization using only one station. We are presenting here the insights of what makes the difference in developing a deep learning algorithm by performing an extensive number of tests. We test the performance of our models in terms of different model parameters, frameworks, and data sets. Moreover, our pipeline is robust and we only work with the raw data. The pipeline is developed and applied at the AQU station placed in the city of L'Aquila in the Abruzzo region (Central Italy). We show that we are able to design a deep learning framework which can be used to extend the existing sparse earthquake catalog of the region in question. Our model is detecting the earthquakes among the random noise with the accuracy of 97%, it is able to determine the events that are close to the station (<10 km) with 94% accuracy as well as identify their belonging to four magnitude classes with a 68% accuracy.

1. Introduction

It is a known fact that some large earthquakes are preceded by foreshocks, events of smaller magnitude that are related in time to the mainshock and located close to its epicenter (Jones & Molnar, 1976, 1979). Their features (e.g., time and space evolution) might be used to predict incoming events; however, seismologist is still not able to prospectively distinguish them from the regular background seismicity (de Arcangelis et al., 2016). Retrospective studies on foreshocks give us insights into physical processes and mechanism governing the earthquake nucleation (Bouchon et al., 2011; Ellsworth, 2019; Ellsworth & Bulut, 2018) and are usually concentrated within few months to days before the mainshock. Yet, many studies revealed that precursory phase might be lasting even longer (Bouchon et al., 2013; Kanamori, 1981; Mignan, 2011; Socquet et al., 2017). To study any length of the earthquake precursor phase, the main source of information is encoded in the earthquake catalogs (Aki & Richards, 2002). Therefore, there is a general interest in extending earthquake catalogs, particularly in reducing their magnitude of completeness (Brodsky, 2019; Lubbers et al., 2018). However, building catalogs with standard seismological approaches is a challenging work,

requiring (i) a network of stations that has been recording at the same place for a long period of time and (ii) a data processing method that is able to scan a large volume of data and produce a practical information about earthquake occurrence.

In the last decades, there have been important advances in automated algorithms to process seismological data for earthquakes detection or phase-picking. Some widely used methods are the short-term average/long-term average (STA/LTA) method (Allen, 1982), the high-order statistics approach such as kurtosis and skewness (Küperkoch et al., 2010; Saragiotis et al., 2002), and the template matching method based on the waveform similarities (Beaucé et al., 2018; Gibbons & Ringdal, 2006; Sánchez-Reyes et al., 2020). Template matching, that performs a scan of the continuous data with characteristic preidentified seismic waveforms, has proven to be quite successful at revealing small noncataloged earthquakes (Kato et al., 2012; Peng & Zhao, 2009; Ross et al., 2017; Shelly et al., 2007). However, this method relies on cross-correlating every preidentified earthquake waveform with the continuous signal, which is highly time consuming when increasing the size and the number of waveforms. Moreover, only the events having a close signature with the preidentified ones are detected. More recently, some data mining algorithms emerged such as the Fingerprint and Similarity Thresholding algorithm (Bergen & Beroza, 2018; Yoon et al., 2015), which does not need a prior knowledge on the existing seismicity. Besides, new methods for features extraction in combination with clustering technique have proven to be quite successful in blindly recognize patterns prior to earthquakes (Seydoux et al., 2020; Shi et al., 2020).

Recently, simultaneous detection and characterization of the earthquake waveforms have been addressed with machine learning algorithms (Bergen et al., 2019; Kong et al., 2019; Mignan & Broccardo, 2020, and references therein). From the range of different artificial neural network (NN) methods, deep learning techniques happen to be the most promising (Mignan & Broccardo, 2020). Specifically, convolutional neural networks (CNN) are suited for image and/or time series data as they are: (i) able to capture spatially or temporally coherent features at different levels, (ii) computationally efficient thanks to parameter sharing, and (iii) necessitate minimal preprocessing since the feature extraction is performed by the network itself. However, they need an extensive amount of data to perform well.

Implementation of deep learning methods within seismological community is numerous. For example, Wu et al. (2018) developed a NN for the detection of geysers' eruptions reaching very high precision/recall rate (0.889/0.923). Furthermore, Perol et al. (2018) developed a CNN for the detection of seismic events and their localization between six geographical regions with a good probability. Using their CNN framework ConvNetQuake they improved the total number of cataloged events by a factor of 17. Lomax et al. (2019) also applied ConvNetQuake to quantify the earthquake detection and characterization on single-station waveforms for worldwide events. Their CNN outputs probabilities for distance, azimuth, depth, and magnitude of the event. Similarly, Magrini et al. (2020) employed ConvNetQuake to quantify the earthquake detection for local worldwide earthquakes (having epicentral distance <130 km). They showed that CNN is capable of detecting events from the regions that were not used in the training data set. Zhu and Beroza (2018) developed PhaseNet, based on U-net CNN architecture, that successfully picks both P and S arrival times, while Zhu et al. (2019) developed a CNN that decompose the input data into signal of interest and noise by learning the representation of data in the time-frequency domain. Wu et al. (2019) developed DeepDetect CNN for detecting start and end times of events of various durations. Ross et al. (2019) trained a recurrent neural network for a seismic phase association within the seismic network. Mousavi, Zhu, et al. (2019) developed an earthquake detector using a fusion between CNN and recurrent neural network units, that outperformed the common existing network architectures. Lastly, Mousavi et al. (2020) trained an earthquake detector and phase picker using complex multitask network structure for decoder encoder model. Their model is applicable to continuous data, the P and S phases precision picking provided by the model is close to the manual human picking, and it is able to detect and pick small-magnitude events.

Besides solving a detection task, other models have been developed for magnitude prediction (Mousavi & Beroza, 2020b), earthquake location prediction (Kriegerowski et al., 2018; van den Ende & Ampuero, 2020) and simultaneously magnitude and location prediction (Mousavi & Beroza, 2020a). However, these algorithms have two constraints, since they either work with a network of stations or they require some additional preprocessing of the input values they are predicting, meaning that they do not use the raw earthquake waveform for the training and the eventual prediction.

Despite all these promising studies, none of them provide a general guideline how to design a good CNN model. It is well known that defining CNN is a tremendously iterative process in terms of hyperparameters (related to the network architecture and the training process) tuning. Moreover, due to its black-box nature, it is difficult to know how different model choices affect the interpretation of final results. To the best of our knowledge, no previous study provides an answer on how to extend near-fault catalog of small-magnitude events when having (i) only one station, (ii) a sparse database of events, and (iii) an extensive amount of recorded data. Therefore, in this study, we address specifically these issues. We develop and apply an end-to-end two-stage pipeline using 1D CNNs for the earthquake detection, localization, and characterization from single-station three-component waveforms. In that context, we provide details of what makes a difference when defining, training, and applying CNN models. For that purpose, we conduct an extensive hyperparameter grid search. We tackle the question of optimal number of label classes, and explore the importance of training data sets as well as the CNN design architectures. This eventually brought us to train and validate 384 different CNN models. Finally, we also provide a methodological pipeline on how one can extend a database of earthquakes using our two-stage pipeline on continuous data.

Our methodology is applied on a region that hosted a MW 6.3 earthquake that occurred on April 6, 2009 (01:32 UTC) near the city of L'Aquila in the Abruzzo region in the Central Apennines of Italy (see Figure 1c) (Sugan et al., 2014; Vuan et al., 2018). This seismic event was preceded by a long sequence of foreshocks and followed by long sequence of aftershocks. Before the 2009 earthquake, the background seismic activity of the area around the epicenter was sparse and mostly clustered in space and time, while the events occurring close to the known faults zone were rare (Bagh et al., 2007; Castello et al., 2006; Chiarabba et al., 2005; De Luca et al., 2009; Tertulliani et al., 2009). Despite a sparse historical seismicity in this region, a seismic station AQU (42.354° , 13.405°) (see Figure 1c), located in city of L'Aquila, has been recording for an extensive period of 29 years, 19 years before the L'Aquila 2009 event. This provides us with a large amount of data to design a framework for a near-fault catalog extension using single-station waveforms.

The paper is organized as follows: first, we explain how we generate training, validation, and evaluation data sets; second, we introduce the two-stage pipeline framework and we discuss different training processes relying on the hyperparameter grid search; third, we evaluate the best CNN models on the unseen evaluation data set and explore the application of the two-stage pipeline on the continuous data. We finish with the discussions and perspectives for future works.

2. Data—Generating Training, Validation, and Evaluation Data Sets

There are two types of data used in this study: the 29-years-long continuous three-component (E-W, N-S, Z) seismograms recorded at the AQU station (see Figure 1c) with the sampling rate of 20 Hz and the existing cataloged earthquakes waveforms (further we denote earthquakes as EQs). The primary goal of this study is to detect and characterize local events in the continuous data. To successfully perform in both tasks, the algorithm has to be able to recognize an EQ event window (positive sample) from a noise window (negative sample). Moreover, it has to learn to correctly characterize the local EQs, e.g., in our case EQs that are in a 10-km radius zone centered in the AQU station. Even though we are only interested into detecting local events our CNN will come across broad examples of EQs within the continuous data. Therefore, we decide to train our model using the extensive database of EQ consisting of local, regional, and teleseismic events to obtain a robust signal detector that is capable of recognizing the general seismic signals buried in the noise data. To further assist into recognizing local events, we choose to train on small time windows of 25-s length, that contain the P and S wave arrivals for the local events. In the case of regional or teleseismic events, these short windows would commonly contain only the P wave arrivals, thus we generate positive samples that are highly distinctive in terms of the waveform shapes. Eventually, our training, validation, and evaluation data sets consist of equally represented 25-s long positive and negative window samples, where positive window samples start 5 s before the catalogued P wave arrivals.

Generating positive samples is achieved by using EQ catalogs. We build our EQ database using three catalogs: the Istituto Nazionale di Geofisica e Vulcanologia (INGV) catalog (INGV Seismological Data Centre, 2006), the United States Geological Survey (USGS) catalog (U.S. Geological Survey, 2017), and the Valoroso catalog (Valoroso et al., 2013). We download the INGV catalog from January 1990 until October 2019

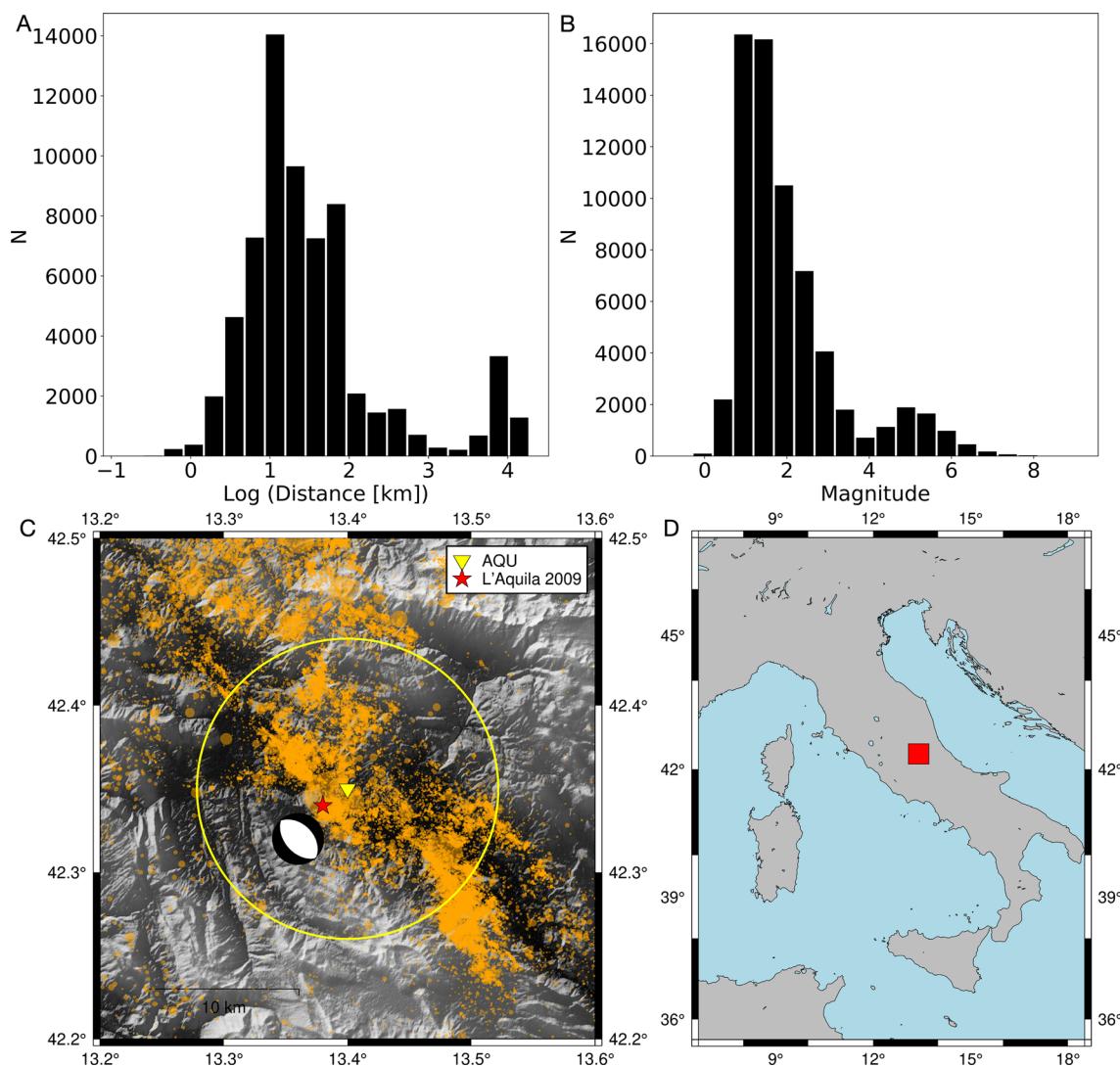


Figure 1. Distributions of events' logarithmic distance (a) and magnitude (b) for a main database denoted as AQULO. Map (c) of the studied area around AQU (42.354° , 13.405°) station located in the city of L'Aquila with the seismicity associated with AQULO database together with the epicenter and focal mechanisms (from INGV catalog) of MW 6.3 earthquake that occurred on April 6, 2009 (01:32 UTC). In the same figure, the yellow circle represents the 10-km radius around AQU station. Map of Italy (d) with the insert of the studied area.

which approximately contains 300 k events. When considering all events that belong to the 10-km radius zone centered in the AQU station we find only approximately 9 k EQs, and only 125 during the period before year 2009. Also, this catalog lacks higher magnitude teleseismic events in the earlier years, therefore to fill this gap we download worldwide EQs with $M > 4.5$ for the same time period, which gives us ~ 179 k events. To reinforce our local events database, we additionally use the Valoroso catalog. This 1-year long catalog for the L'Aquila 2009 event is the largest event database for a moderate magnitude normal faulting EQ. There are ~ 64 k events generated by an automatic P and S wave picking procedures. It spatially spans the area of the two major faults: the high-angle L'Aquila fault and the listric Campotosto fault.

The three catalogs are merged, the duplicate events are removed and for each event we calculate the arrival of P waves at the AQU station. Finally, we end up with ≈ 564 k cataloged EQ events. From this extensive list, the positive samples retained in our data set are chosen by satisfying several conditions. First, we keep only the window frames containing one unique event; second, all three components have to be present (82% of the EQ events); third, the signal-to-noise (SNR) threshold must be larger than 2 (32% of the EQ events). The SNR is calculated by computing the ratio of the signal and noise standard deviations. The positive samples

having one or more components satisfying the SNR threshold are accepted. Applying these three conditions we obtain 65,865 positive samples. In Figure 1, we show the epicentral distance (A) and the magnitude (B) histograms of the selected EQ events as well as the studied area (C). The complementary negative samples are random noise time windows selected between the cataloged ones. They are sampled equally during the night and day-time period as well as over the years. We do not apply any additional constraints on the negative samples; therefore, they truly represent random seismological signals of various amplitudes. We generate the same number of the negative samples as the positive ones.

Each sample is detrended and normalized with the maximum value out of the three components. Thus, we only keep the information about the EQ waveform shape while the information about the amplitude is lost between the individual positive samples. We do not perform any type of filtering leaving to the network to extract the best features by itself. While the negative samples have a uniform distribution over the years, this is not the case for the positive samples. Around 40% of positive samples are associated with the Valorooso catalog during the 2009 years. From Figure 1, we notice that the database is imbalanced in terms of distance to station and magnitude. This distribution is expected by the Gutenberg-Richter law (Gutenberg & Richter, 1955). Furthermore, since our data covers a long period of 29 years, the instrument sensor has been changed over the years, which may cause varying transfer function. However, these variations are partly erased by the normalization. We refer to this original database further as AQULO.

3. Methodology, Training, and Validation

In this study, we develop a two-stage pipeline consisting of two individual 1D CNNs: first, a detector CNN that learns to classify the EQ windows from the noise windows; second, a characterization CNN that performs two classifications on the detected EQ samples simultaneously: the epicentral distance and the magnitude (see Figure 2a). In both stages, we are solving classification problems, but compared to what have been done so far in literature, they are separated (Lomax et al., 2019; Magrini et al., 2020; Perol et al., 2018). We chose to separate these problems since we expect that CNNs should learn different features when identifying earthquake from noise waveforms (first stage) compared to the situation when it learns only from earthquake waveforms (second stage). Furthermore, the two stages have different proportions of samples in each class, in the training set and also once applied on real continuous data. The first stage is trained using balanced data set, since we generate the same number of negative as there are positive samples: this helps the model to learn a better classification. But in reality once applied on the continuous data the problem is highly unbalanced as very few events happen with respect to the noise. The second stage is trained using unbalanced data set, since it is too difficult to have the same number of events belonging to different distance and/or magnitude classes due to natural earthquakes properties (e.g., magnitude distribution). Once applied on continuous data this model is only exposed to detected events in the first stage.

To reach the full potential of CNN modeling, for detection and characterization CNNs, we perform a grid search over the main hyperparameters explained and listed in S1 (Supporting Information) (Ismail Fawaz et al., 2019). Additionally to commonly used hyperparameters, we also test two different CNN architectures (see Figure 2b), where the seven-layer one is similar to the architecture developed in Perol et al. (2018). Two different architectures are tested for both stages of our pipeline. Implementation details are explained in the Appendix. In total, the combination of all tested hyperparameters, lead to the grid search that consists of 48 tests. In machine learning developing, a model is performed in three different stages that are associated with three different data sets. The first one is a training data set for which one fits the model. The fitting stands for minimizing the residual between the model prediction and the ground truth. The validation set is used to see how well the model is generalizing, as well as to select the best hyperparameters from the grid search. The third data set, the evaluation or test data set, is an unseen data set that is used to obtain the final evaluation of the best model, in order to assess that no overfitting or bias was performed during the grid search. For this purpose, our original data set, named AQULO, is split into three data sets in proportions: 80%, 10%, and 10% for training, validation, and evaluation, respectively.

Next, we present the validation results for the detector CNN grid search (first stage), as well as the sensitivity of the characterization CNN grid search (second stage) for different multiclass definitions, data sets,

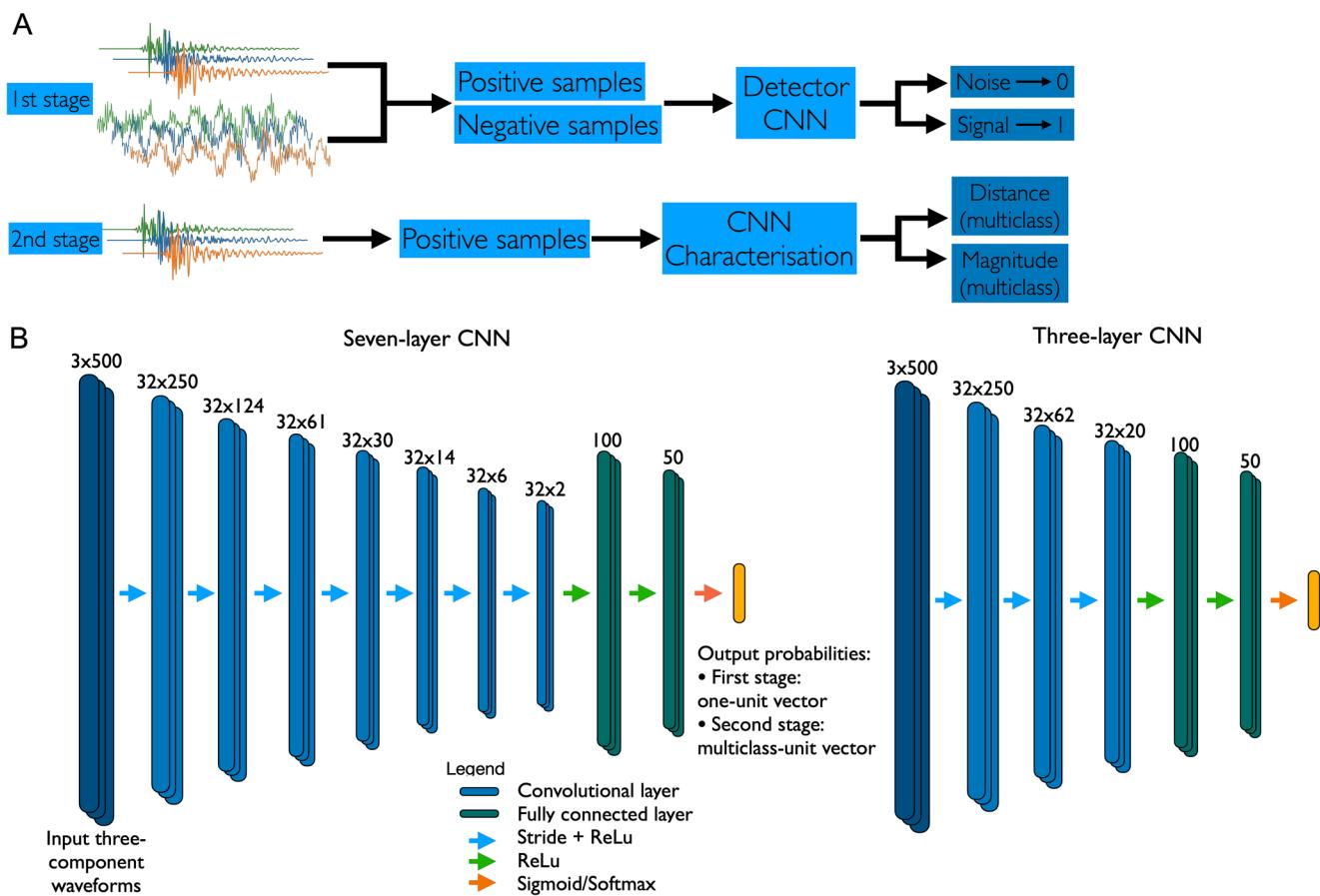


Figure 2. (a) Schematic representation of the two-stage pipeline using 1D convolutional neural networks (CNNs) for earthquake detection (first stage) and characterization (second stage). (b) The network architecture for seven-layer CNN and three-layer CNN models. The input is the 25-s long three-component waveforms sampled at 20 Hz; hence, the input layer has dimension 3×500 . The output of the CNN is either one-element vector (the first stage) or multiclass-element vector (the second stage). In first stage, one-element vector tends to zero for negative samples (noise) and it tends to one for positive samples (EQ). In the second stage, the output probability vector depends on the number of distance and magnitude classes. When the CNN model is more confident than the input sample belongs in a particular distance/magnitude class, then the probability associated with this particular unit of the output vector is closer to one. Between input layer and output probabilities, there are seven-convolutional and three-convolutional layers and two fully connected layers. The operations between layers are represented by arrows and their meaning in the legend. The downsampling is done by 1D convolution and stride, where the length of convolution kernel is set to three points and stride is either set to two data points for all layers (seven-layer CNN) or two-four-three data points (three-layer CNN).

and different problem definitions. The performance of different models is quantified by the accuracy value (see S2 in Supporting Information for more details on the used statistical metric).

3.1. First Stage: Hyperparameter Grid Search Results

The first stage, distinguishing the EQ window versus the noise window, has been investigated by several studies in the last years (Lomax et al., 2019; Magrini et al., 2020; Perol et al., 2018). The grid search with the accuracy values for each 48 test is summarized in Figure 3a. Most tests reach high accuracy of 96%–97%, while tests that reach only 50% accuracy are always associated with the SGD optimizing algorithm and low values of the learning rate (see S1 in Supporting Information for the parameter explanation). The test with the highest value is printed with tilted, orange font, and the associated hyperparameters is listed in Table 1 under Experiment 1. The corresponding learning curve (see Figure S1a) shows a relatively rapid learning, that reaches stable validation loss after 169 epochs, while also showing a loss drop at epoch 50. The relation between the training and validation losses indicate a good fit. We additionally plot the accuracy curves for training and validation in Figure S1b.

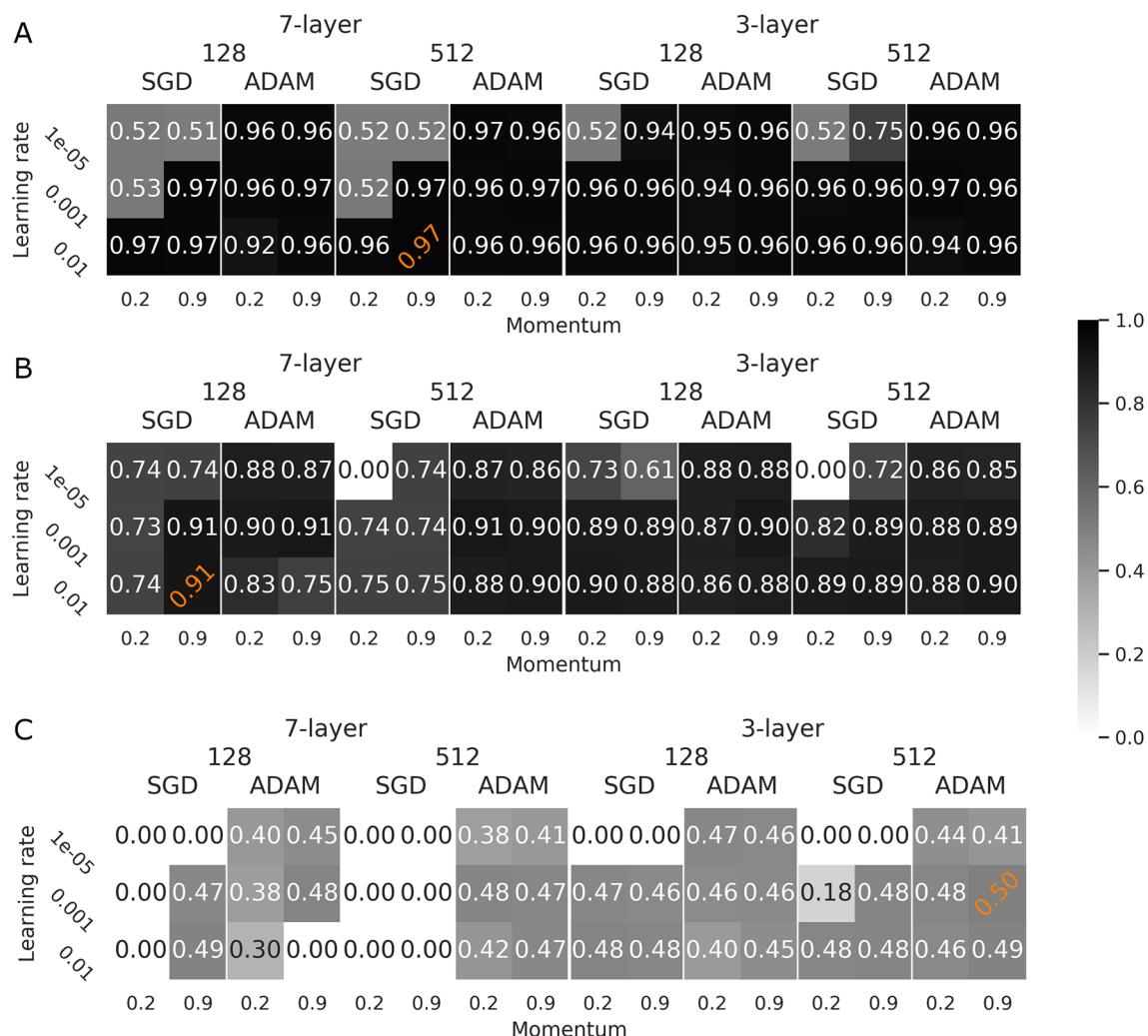


Figure 3. (a) Grid search result for a convolutional neural network (CNN) detector represented by maximum scored accuracy values. (b) Grid search result for a CNN characterization and a distance classification where tested distance configuration has two classes (<10, >10 km). (c) Same as (b) but for a magnitude classification where tested magnitude configuration has four classes (<1, 1–2, 2–3, >3). Tests with the highest scored accuracy for each panel are printed with tilted and orange fonts.

Table 1

List of Eight CNN Experiments Associated With a Detector or Characterization Model and Different Databases Together With the Associated Hyperparameters and Scored Accuracy Values for a Distance or Magnitude Classification Where AR, O.A., L.R., M.R., D.A., M.A. Stands for an Architecture, an Optimization Algorithm, a Learning Rate, a Momentum Rate, a Distance Accuracy, and a Magnitude Accuracy, Respectively

Exp.	Stage	Task	Database	AR	O.A.	L.R.	M.R.	Batch	D.A. (%)	M.A. (%)
1	First	EQ, noise	AQULO	Seven-layer	SGD	1e-2	0.9	512	—	—
2	Second	EP2M2	AQULO	Seven-layer	ADAM	1e-3	0.2	512	91	93
3	Second	EP2M4	AQULO	Seven-layer	SGD	1e-2	0.9	128	91	49
4	Second	EP2M4	AQULO_MB	Three-layer	ADAM	1e-3	0.9	512	90	61
5	Second	EP2M4	AQULO_RMS	Three-layer	ADAM	1e-3	0.2	512	92	53
6	Second	EP2M4	AQULO_FILT	Seven-layer	ADAM	1e-3	0.2	512	90	54
7	Second	EP2	AQULO	Seven-layer	ADAM	1e-3	0.2	512	91	—
8	Second	M4	AQULO	Three-layer	SGD	1e-2	0.2	128	—	53

3.2. Second Stage: Hyperparameter Grid Search Results and Sensitivity Study on the Multiclass Definition

The second stage, the characterization CNN model, is a more complex classification task since intrinsically it is difficult to distinguish between EQ events that have close distance and magnitude values. Lomax et al. (2019) show that their CNN model encountered difficulties in learning the EQs' characteristics (distance, magnitude, depth, and azimuth) because the training density is low and the data set contains the worldwide events, which have a large variability. In our case, the goal is specifically defined, we seek to correctly characterize EQ events that belong to the near-fault seismicity. For this purpose, the distance task is more important than the magnitude one. First, we separated our data in only two classes for each task (distance and magnitude). For the epicentral distance, this means separating EQs closer to 10 km (<10) from the station from the rest (>10). By setting the boundary at 10 km, we focus our attention to small magnitude EQs that are happening close to the station, meaning close to the fault. For the magnitude, the boundary is set to MW 4. We refer to this characterization model as EP2M2. For the EP2M2, the grid search reaches the validation accuracy between 75%–91% and 89%–94% for the epicentral distance and magnitude, respectively (see Figure S2). The hyperparameters associated with the best model are listed in Table 1 under Experiment 2. The learning and accuracy curve for the best model are shown in Figure S3. Overall, we notice that the CNN characterization model successfully performs classification by the distance and the magnitude classes, even though the positive samples used for training are normalized. This indicates that distance as well as the magnitude could be estimated just from the earthquake waveform shape as has already been applied in other methods (Rodríguez-Pradilla & Eaton, 2019).

Although the EP2M2 configuration gives us satisfactory results, to be more precise in characterizing local events we also test a magnitude classification with four classes, specifically: <1 , $1\text{--}2$, $2\text{--}3$, >3 . We call this new characterization model EP2M4. Achieving a good accuracy is harder since statistically a correct random guess would drop from 50% chances in the 2-class problem to 25% in the 4-class problem. The grid search results for this EP2M4 model are summarized in Figures 3b and 3c. The distance prediction overall stays the same as in the previous configuration, while the magnitude prediction reaches in average the accuracy of 45%. We observe that most tests for the distance classification reach satisfactory accuracy, unlike for the magnitude classification where 30% of the tests, mostly associated with seven-layer CNN coupled with SGD optimization algorithm, are not able to converge.

Furthermore, the best test for the distance classification does not match the best test for the magnitude classification. In this study, the priority is the distance prediction over magnitude. Based on these considerations and on the learning curve, the hyperparameters for the best model are listed in Table 1 under Experiment 3. The learning and accuracy curve for the best model are shown in Figure S4.

3.3. Second Stage: Sensitivity Study on the Database Selection Process

In this section, we study the sensitivity of the characterization model to the database selection process. We focus on the fact that the AQULO database for the EP2M4 is highly unbalanced between the classes of the two tasks (see Figures 1a and 1b) while a CNN training is usually best performing with balanced classes. To tackle this problem, we refine our database AQULO, so that each of the four magnitude classes have the same number of examples. In the classes with extra positive samples, we keep the samples having an SNR close to the average SNR value, therefore removing samples with extremely low or high SNR values. We further on refer to this database as AQULO_MB. Eventually, the AQULO_MB database is the magnitude balanced subset of the AQULO database. In total, we lose 50% of the original training and validation database. The grid search for this experiment provides us with several models where the magnitude accuracy is higher than the original EP2M4 experiment (see associated grid search values in Figure S5). However, the distance accuracy does not improve. It overall deteriorates, since many tests have an accuracy equal to 0 which shows a less robust model. The hyperparameters for the best model are listed in Table 1 under Experiment 4, and the learning and accuracy curve for the best model are shown in Figure S6.

By using a relatively low SNR threshold for selecting events in Section 2, we choose many noisy positive samples. Yet, this enables us to build a large and diverse database of earthquakes waveforms. We will now investigate how the model performance is affected by a more careful selection process of the positive

samples at the expense of losing some of them. Therefore, we generate two more databases. First, for database named AQULO_RMS, we recalculate the SNR by using the root mean square (RMS) of the signal over the noise envelope and further on we accept only those positive samples reaching $\text{SNR} > 5$ for all three components. Hence, we clean AQULO by losing 74% of the positive samples. For the second database, named AQULO_FILT, we just filter AQULO by keeping only the positive samples with $\text{SNR} > 3$ and distance $< 200 \text{ km}$. This approach reduces the original database by 40%. All generated databases are listed in Table S2 with the total number of events within each one, mean and standard deviation of the logarithmic SNR values associated with samples within each data set and the selection rules. When we compare four different databases in terms of the average SNR value from the highest to the lowest we get AQULO_RMS, AQULO_FILT, AQULO_MB, and AQULO, respectively. Additionally, we classify these databases in eight groups associated with all combination of distance and magnitude classes (see Figure S7). We notice that applying more rigorous thresholds mostly affects events associated with having distance $> 10 \text{ km}$.

Running the grid search for the AQULO_RMS and the AQULO_FILT databases for the EP2M4 configuration gives slightly better distance as well as magnitude accuracy values compared to Figures 3b and 3c, however not larger than 1%–3% in total (grid search for AQULO_RMS and AQULO_FILT is shown in Figures S8 and S10, respectively). We can draw two important conclusions from these three training experiments: first, training distance and magnitude classification models on smaller databases can perform as well as on larger ones; second, the databases with clearer positive samples (with a higher mean SNR) do not provide significantly better results. The hyperparameters associated with the best model for AQULO_RMS database are listed in Table 1 under Experiment 5. The learning and accuracy curve for this model are shown in Figure S9. Similarly, the hyperparameters for AQULO_FILT database are listed in Table 1 under Experiment 6. The learning and accuracy curve for this model are shown in Figure S11.

3.4. Second Stage: Sensitivity Study on the Problem Definition

In this last experiment, we assess the distance and magnitude classification by separating the second stage into two individual tasks, meaning that we train a CNN model for each task separately. The grid search results for distance classification are shown in Figure S12 and for magnitude classification in Figure S14. Both experiments do not generate any result that is significantly better to ones in Figures 3b and 3c for EP2M4 configuration. The hyperparameters associated with the best model for distance classification are listed in Table 1 under Experiment 7. This model is not chosen considering the foremost distance accuracy value, but on the shape and the convergence of the training and validation learning curves. The learning and accuracy curve for this model are shown in Figure S13. Similarly, the hyperparameter for separated magnitude classification is listed in Table 1 under Experiment 8. The learning and accuracy curve for this model are shown in Figure S15.

3.5. Summary of all Experiments

In Table 1, we summarize all experiments with the hyperparameters associated with the best models for each experiments, selected in previous subsections on the scored accuracy values and learning curves. Additionally, in the same table, we list the reached distance and magnitude accuracy values for the experiments associated with the CNN characterization model (from second to eighth experiment in Table 1). We notice that for each experiment the best model is associated with a different combination of hyperparameters depending on the task or the database used. This demonstrates that a grid search should always be performed. We also observe that the SGD in combination with the seven-layer architecture usually fails in the training as well as the SGD with as small learning rate of $1\text{e}-3$. However, we can see that some hyperparameters choices seems to be more robust across the experiments: a seven-layer architecture, the ADAM optimizer, a moderate learning rate of $1\text{e}-3$, a momentum of 0.2, and a batch size of 512. If we select the aforementioned hyperparameters as a default one for all experiments, the accuracy difference is always within 5%. Since in all individual experiments, the distance classification does not score higher than 92% we reach the maximum capacity with this specific training and validation database and our CNN models (see column D.A. in Table 1).

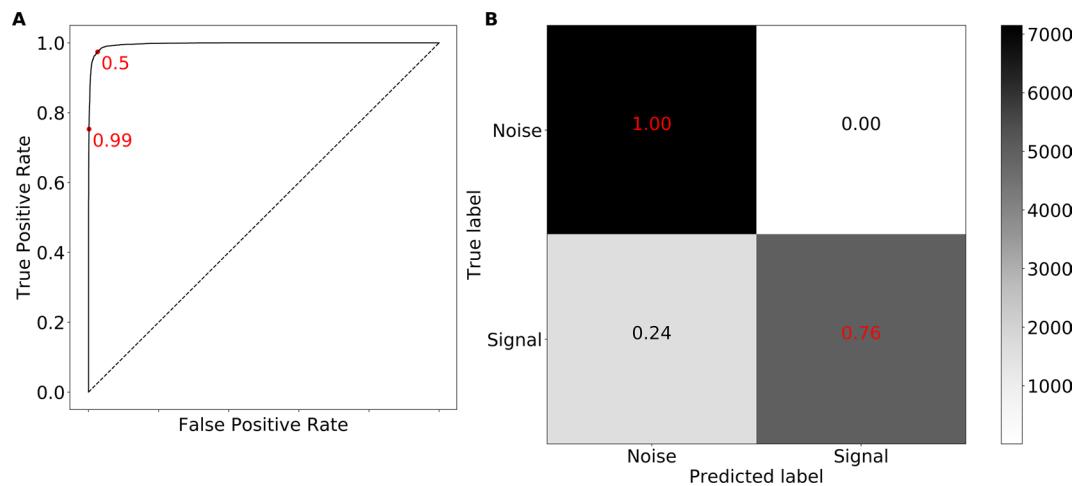


Figure 4. Receiver operating characteristic curve with two indicated confidence threshold for the best detector convolutional neural network (CNN) (a) associated with hyperparameters listed in Table 1 under Experiment 1. Confusion matrix (b) for the same detector model when the confidence threshold is set to 0.99.

We implement the whole pipeline using PyTorch package (Paszke et al., 2019). The training is performed using NVIDIA Tesla V100 graphics processing. The training time is scalable with the number of epochs and 5,000 epochs takes ~3h30. More implementation details can be found in the Appendix, while grid search results and associated learning and accuracy curves for the best models in the Supporting Information.

4. Application

The detector and characterization CNN models that performed the best in the experiments of Section 3 (summarized in Table 1) are further tested on the evaluation data set in order to avoid any overfitting on the validation data set used for the grid search. This evaluation database is the same for all experiments, except for AQULO_RMS and AQULO_FILT databases.

Next, we first present the results for the evaluation of detector CNN, then, we compare the performance of seven characterization CNN models on the evaluation data set. Finally, the best detector model and characterization model are applied on the continuous data.

4.1. Earthquake Detection

For the best detector CNN model (see Table 1), we calculate the receiver operating characteristic (ROC) curve (Fawcett, 2006) shown in Figure 4a (see S2 in Supporting Information for more details). The ROC curve is calculated on the evaluation data set by varying the confidence threshold on the detector CNN model output, that is the detector probability. We see that our detector is almost a perfect classifier between earthquake and noise windows with the area under the ROC being 0.99. For a confidence threshold set to 0.50, we reach a recall of 97%, a precision of 97% and an accuracy of 97%. Further, by setting the confidence threshold to 0.99, we reach a recall of 75%, a precision 100%, and an accuracy 88%, and the associated confusion matrix is shown in Figure 4b. We notice that by setting the confidence threshold low as 0.50 the misclassified event are associated with two groups: some high magnitude teleseismic events, and some low magnitude regional events (see Figure S16). Both have low SNR, which might explain why they were not detected. However, by taking 0.99 as a confidence threshold, we do not have any False Positives, so even if we missed some events, we are sure that we will have close to zero false detections (see Figure 4b). For our particular application, it is important to have a very high precision since once this model is applied on the continuous data, the number of noise samples is extremely higher than the number of EQ, which could cause many false detections. Further, we explore the relation between the evaluation accuracy values with respect to the SNR values for the positive samples. We form equally sized bins of 1k samples each. As expected the accuracy increases with the SNR, scoring above 99.1% for SNR larger than 0.64 (see Figure S17). We also

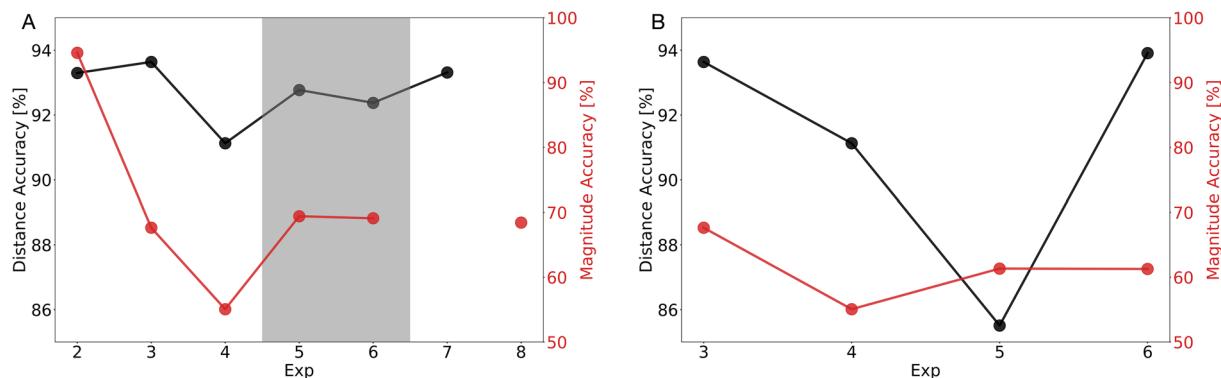


Figure 5. (a) Evaluation accuracy values scored for distance and magnitude classification for different characterization convolutional neural network (CNN) models associate with experiments listed in Table 1. Experiments 5 and 6 within the gray space are associated with AQULO_RMS and AQULO_FILT evaluation databases, respectively. (b) Same as before just for models associated with Experiments 3–6 from Table 1 where all models are evaluated on the same AQULO evaluation data set.

explore how the mean detector probability of the model for the positive samples, reflecting the confidence of the model, depends on the SNR value. While the same relation can be seen, i.e., a better confidence is attributed to high SNR samples, we can also point out that this confidence stays very high (see Figure S17).

4.2. Earthquake Characterization

For the best characterization CNN models (experiments from 2 to 8 in Table 1), the scored distance and magnitude accuracy values on the evaluation data set are shown in Figure 5. In Figure 5a, the nonshadowed area is associated with models tested using AQULO evaluation database and the shadowed is associated with AQULO_RMS or AQULO_FILT ones. We first notice that the values obtained for the evaluation data set are comparable to the ones obtained on the validation data set, which indicate a good generalization of our models. Next, we observe that the characterization CNN model having the highest distance accuracy is associated with the model denoted as Experiment 3 in Table 1. This model corresponds to the 2-distance and 4-magnitude task and is trained on the AQULO database. Considering the validation accuracy values in Table 1, this model is the second best one when predicting the distance class, being surpassed by the model trained with AQULO_RMS database. However, it has to be taken into consideration that the AQULO_RMS has been cleaned of all noisy positive samples by using a high SNR during the selection process. The magnitude classification for the 4-magnitude task is well performed by four models associated with Experiments 3, 5, 6, and 8 (see Table 1). Heretofore, we compare CNN characterization models by evaluating them on different evaluation databases to assess information of distance and magnitude accuracy. Further, we compare the models associated with Experiments 3, 4, 5, and 6 by calculating their performance on the same evaluation database (AQULO) shown in Figure 5b, in which, the distance and magnitude accuracy values for Experiments 3 and 4 are the same as in Figure 5a. It is important to note that the model associated with Experiment 3 is having the highest distance and magnitude accuracy values.

Since, in this study, we give advantage to distance classification over magnitude, we choose as a definitive characterization CNN model the one trained with the AQULO database for the EP2M4 task (Experiment 3 in Table 1). Its confusion matrix is shown in Figure 6 and it is obtained by considering that the correct distance and magnitude class is the one with the highest probability value for the distance and magnitude classification, respectively. It reaches a distance classification accuracy of 94% and magnitude classification accuracy of 68%. From Figure 1, it is clear that the AQULO database is unbalanced, therefore we notice that our model gives a better accuracy for the distance class >10 km and magnitude class 1 < M < 2. Eventually, we are interested in extracting local events corresponding to class <10 km; therefore, it is desirable that far distance events are labeled as local ones in <3% (see Figure 6). If we only consider local events and again calculate the magnitude classification accuracy, we reach an overall magnitude accuracy of 73%. Overall, we notice that the earthquake samples (the positive samples) are mislabeled when their labels are close to the classes' boundaries, e.g., see Figures S18a and S18b (in Figure S18a all events below $\log(\text{Distance}[\text{km}]) = 1$

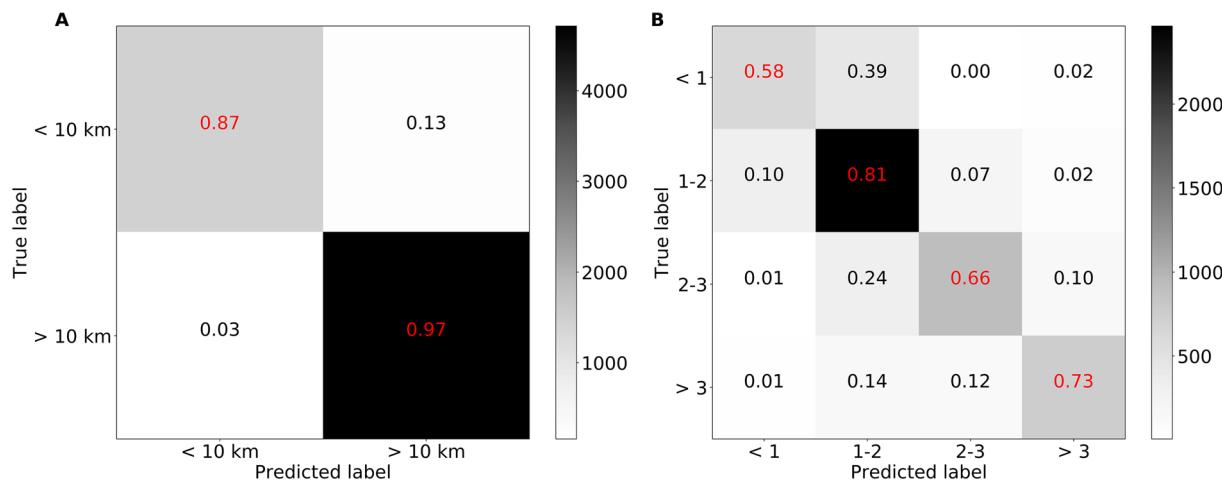


Figure 6. Confusion matrices for distance (a) and magnitude (b) classifications associated the characterization convolutional neural network (CNN) model listed under Experiment 3 in Table 1.

are classified as being nonlocal when actually they are local and vice versa) for the distance and magnitude classification. This is expected, since intrinsically it is harder to distinguish between EQ events that have very similar magnitudes and distances.

Next, for our chosen characterization CNN model, we calculate the distance and magnitude accuracy with respect to the SNR value (see Figure S19a). The expected behavior would be an accuracy increasing with the SNR value. While this is mostly true for the distance, the magnitude classification has the highest score for the [0.84, 1.14] bin, the second highest bin. We can further look at the mean probability values with respect to the SNR value (see Figure S19b). While for the magnitude classification higher SNR means also better model prediction, for the distance classification the lowest SNR group reaches the foremost probability values (Mousavi et al., 2020). The SNR of the events is not proportional to the output CNN probability, most probably because this probability is product of nonlinearity introduced within the algorithm. Thus, it is difficult to directly interpret and it remains an open question. The events that belong to the lowest SNR group follow the same distance and magnitude distribution as in Figures 1a and 1b, so their distribution is a consequence of the Gutenberg-Richter law.

As for the validation data set, we notice here that one can reach good results with a smaller database if the samples have high SNR values, such as the AQULO_RMS. The same conclusion stands when comparing results for AQULO_MB and AQULO_FILT: the two sets have similar number of events, but the results for AQULO_FILT outperform AQULO_MB results. The reason is because AQULO_FILT has positive samples that in total have higher SNR values than the ones belonging to AQULO_MB (see Table S2).

Overall, the evaluation results for the characterization CNN model are satisfactory, since applying it on the continuous data we are confident that newly detected events would be properly labeled as the ones corresponding to the near-fault location.

4.3. Application on the Continuous Data

Based on the validation results and on the evaluation made on the unseen data, we show that the CNN models from the Experiments 1 and 3 (see Table 1) are accurate and generalize well to new data. These two models are further applied on the continuous data.

The three component time series are streamed with a 1-s slide window of 25-s length ($\approx 96\%$ overlap). Furthermore, each 25-s long windows are detrended and normalized with the maximum value of the three components. Processing 1 day of data on a single GPU with our two-stage pipeline takes 2.5 s. Once we obtain the model probabilities for the two CNNs we perform a postprocessing to extract new events. The postprocessing consists of setting up a threshold to the detector output, keeping only a single 25-s window

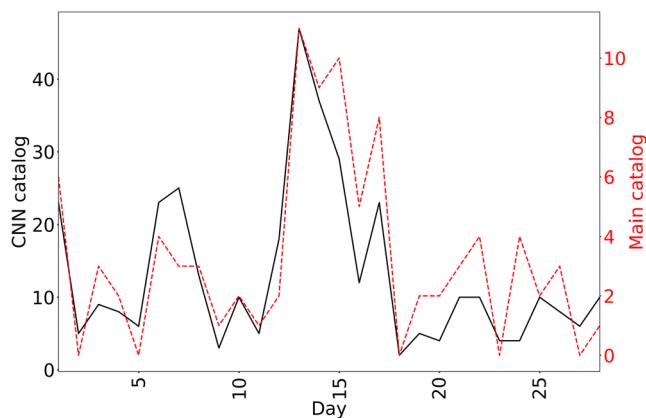


Figure 7. Number of new local detections (within 10 km from the station) per day obtained by two-stage convolutional neural network (CNN) pipeline for February 2009 compared with the existing local catalog.

seismicity trend is following the existing one. From these new 273 detected events, 84% are labeled as having a magnitude lower than 1 and the rest with a magnitude between 1 and 2. In Figures 8a–8c, we see several examples of new detections together with the detector probability values for every sliding window.

Each second represents the prediction for the center of the 25-s sliding window; therefore, we notice when the slide window starts to capture an EQ the probability rises. It reaches the highest probability when the EQ is positioned in the center of the slide window marked with the first dashed line. The second dashed line marks the end of that window. For one event we can have several window samples with high probability since we use sliding with $\approx 96\%$ overlap. We only keep one window for each event associated with the maximum probability value. From Figures 8a–8c, we notice that the CNN really learns to recognize specific waveform shape within the 25-s window frame and all the windows surpassing the P -phase arrival time show a very low detection probability. All 273 new February detections were checked visually and confirmed as being true detections. We also notice that most of these new events are detected between midnight and 6 a.m., when the noise level is the lowest.

Running our detector on more periods and checking the detections, the large majority was correct. Yet, some particular false detections are happening on specific periods of time corresponding to a perturbed recording. By looking closely (see Figures 8d–8f), we can see that the signal shows short gaps in the time series, randomly within the three components (in Figure 8 see E-W (D) and Z (F) components). While we do not perform any preprocessing, the CNN by itself learns to filter the data in the first convolutional layers. Yet, it is to be expected that these small gaps are band-pass filtered at the entry of the model and thus eventually resemble EQ events having very sharp distinctive phase-like features. In order to avoid such false detections, some examples must be included in the training set as negative samples. Since gaps in time series are always manifesting themselves in the same manner, this problem could also be easily solved by first flagging windows that contain gaps. It is nonetheless interesting to see that such signals were detected even though they are not part of the training data set. This is a specificity of the machine learning as compared to template matching, which could be very useful even though it can also lead to unusual misclassifications as can be seen in this example.

5. Discussion

In this study, we developed a two-stage pipeline for earthquake detection and characterization for the purpose of extending a local near-fault catalog. The problem we want to solve is specifically related to our study area, a small region around AQU station that was unexpectedly extremely seismically active in a period around the 2009 L'Aquila earthquake. To develop a pipeline that would successfully reveal new earthquakes in existing single-station three component data, we develop several steps in our study. First, we decide upon

of the same event, extracting distance and magnitude label for each detected window, keeping only those windows that are classified as local ($ep0$) and finally comparing all detected events with the events in the existing EQ catalog. For the latter, a match is obtained if a new detection (a 25-s window) is within the adjustable (e.g., 50-s) window frame around the P -arrival of the cataloged event.

We test our pipeline on a 1-month (February) foreshock period of L'Aquila earthquake in 2009 (Sugan et al., 2014; Vuan et al., 2018). To have the best detector precision, we chose to set a confidence threshold to 0.99 (see Section 4.1). During the analyzed period, we detect in total 369 local events. Out of these detections, 90 events are already in the main catalog. Here, the main catalog is the one obtained in Section 2 by merging INGV, USGS, and Valoroso catalogs. Only one event is missed from the main catalog by our two-stage pipeline, and 273 new local events are found, while 6 events from 369 detections where labeled as local when in reality those are nonlocal events. The total number of detected local events is compared with the existing number of local events in Figure 7. Hence, we detect more events than there are in the main catalog and the new

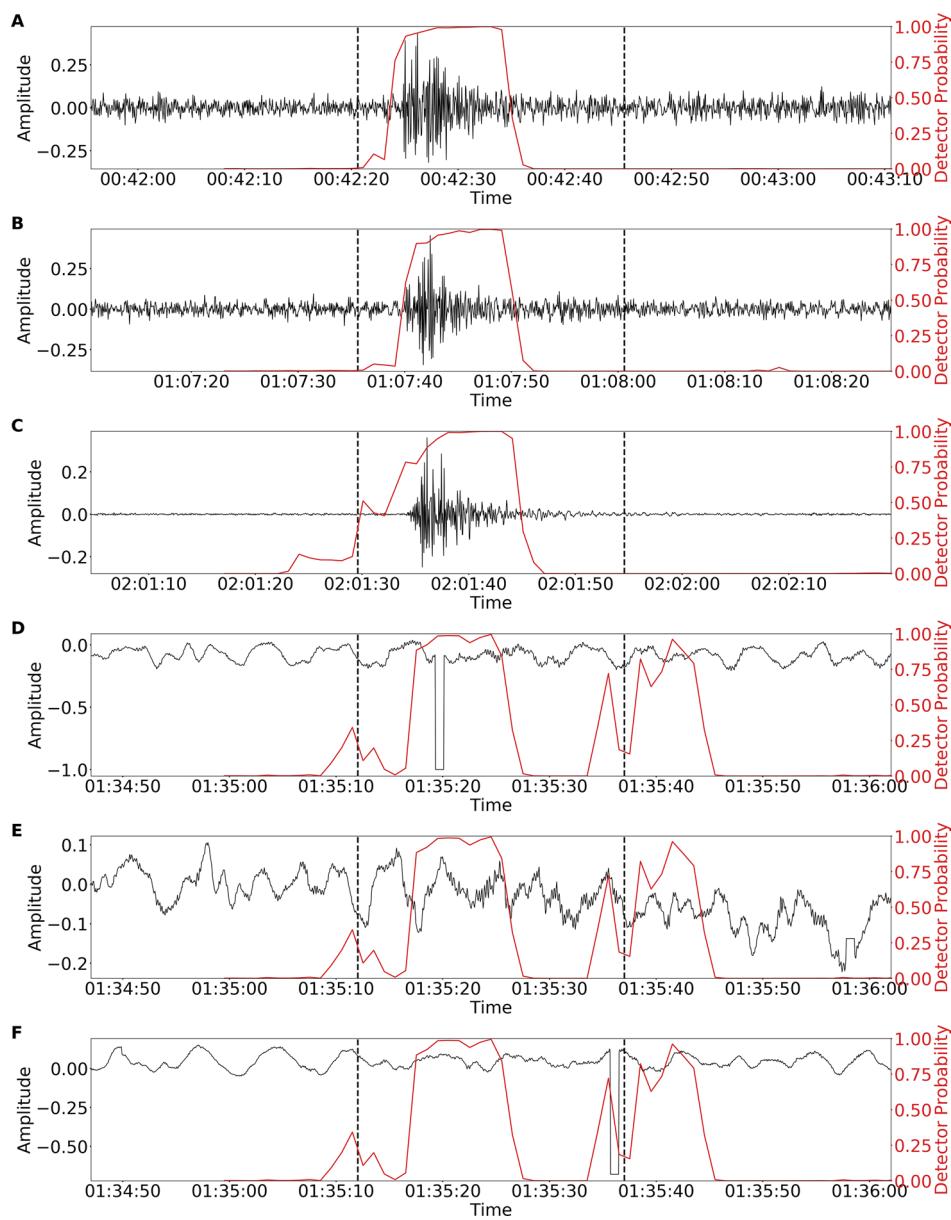


Figure 8. (a–c) Convolutional neural network (CNN) detector probability (right y-axis) together with the time series amplitudes of Z component (left y-axis) for several newly detected local events in February 2009 classified as signals with maximal reached probability, distance class, and magnitude class as (a) 0.999, ep0 (<10 km), m0 (<1), (b) 0.998, ep0 (<10 km), m0 (<1), (c) 1.000, ep0 (<10 km), m1 (1–2). (d–f) CNN detector probability (right y-axis) together with the time series amplitudes for (d) E-W, (e) N-S, and (f) vertical components (left y-axis) for one False Positive detection associated with the existing data gaps in February 1994.

the window size that should emphasize the features of local events we want to detect. Second, we generate an appropriate database of positive and negative samples of the window length chosen in first step. Third, we define an algorithm that corresponds to our need for earthquake detection and characterization, for last selecting epicentral distance and magnitude as appropriate labels. Fourth, we train and validate our two-stage pipeline by simultaneously studying how the hyperparameters, different distance and magnitude classes, dissimilar databases, and a problem definitions affect the final distance and magnitude classification accuracy values. Fifth, we test several models on the evaluation data set. Finally, we apply the best detector and characterization CNN models on the continuous data and extract local near-fault earthquake events.

We find that having trained our CNN characterization model on wide range of different EQs using only 25-s window, gives us capability to successfully detect near-fault events and as well acknowledge the existence of large events in our continuous data. It might be that changing this window could change the performance and/or application of CNNs, a problem that has already been addressed (Dai & MacBeth, 1995).

We further show that choosing CNN hyperparameters does make a difference when training CNN on EQ signals. Therefore, a grid search for any deep learning training is always preferable. Even in case where one is working with large size data set, it is advisable to perform relevant grid-search on a subset of the original data set, to gain an insight about right parameters' value. Also, we saw that there exists a general combination of hyperparameters that will eventually work well for many individual problems for earthquake location and characterization when training CNN model. This combination refers to: the seven-layer CNN architecture, the ADAM optimizer, a $1e-3$ learning rate, a 512 batch size, and a 0.2 momentum. It is also interesting to note that CNNs with the seven-layer architecture overall perform better, meaning that for successful EQ detection and characterization, calculating low-frequency features (using a deeper architecture) is important (Perol et al., 2018). The CNNs are highly parameterized algorithms, and beside the hyperparameters and two different architectures that we tested for, one could also go further and test in more details other specific parameters of the CNN architecture. In that context, a complementary study would be exploring the optimal number of neurons in each layer, the kernel size, the pooling layers, the stride size, the number and type of layers, the dropout regularization, the types of activation functions, etc. Due to the number of different parameters to test and their possible implication on the final results, this study presents a future perspective.

By comparing results obtained for different data sets (see Figure S7), we show that the accuracy of the characterization CNN model depends on the database size as expected: a larger database generates better results. However, we notice that in case of a smaller data set, the characterization CNN accuracy is highly dependent on the database quality. One can reach good results with a smaller database if the samples have high SNR values, as shown in other studies (Mousavi et al., 2020). For regions or stations that have small-scale catalogs, one possible solution is definitely data augmentation (Chatfield et al., 2014; Cui et al., 2016; Le Guennec et al., 2016). The most common data augmentation techniques for a time series are adding artificial gaps to the training samples, the circular rotation (that is randomly shifting the *P*-arrival within the time window of the event) and adding random Gaussian noise (Zhu et al., 2020). Some other solutions when dealing with a sparse seismic catalog might be implementing a generative adversarial network (GAN) (Goodfellow et al., 2014) that is specialized in generating new examples of a given training set within some statistic parameters (Li et al., 2018) or performing transfer learning by retraining the existing model with the positive and negative examples from a different region.

To assess the performance of our developed two-stage pipeline, we compare it with an existing advanced deep learning algorithm in literature (Mousavi et al., 2020) that was trained using a worldwide database of the earthquake and noise examples (Mousavi, Sheng, et al., 2019). We test the two models on 1 day during the foreshock sequence on February 1, 2009. Our pipeline detects 40 events, while EQTransformer detects 27 events, 53% of our events were detected by the EQTransformer, while 78% of the EQTransformer events were detected by our pipeline (see S3 for the implementation details). In total, there is 46% overlap between the two lists of detections. The percentage of the overlap indicates that the two models are complementary. Therefore, this justifies the importance of developing a deep learning model for a specific location. Also, one should keep in mind that the actual detections revealed by the two models highly depend on the utilized probability threshold.

In our experiments, we also show that there is a level of accuracy that one can reach when using the specific EQ database and training CNN for EQ detection and classification. While we can reach good detection accuracy, the characterization process is still challenging. We can argue that while there is enough diversity between EQ signals and noise in general that allows CNN to easily recognize an event-like waveforms, this is only partially true for the distance and magnitude classification. The level of similarity between events that share common distance and magnitude labels is quite high, so separating them arbitrarily in different classes presents a real challenge for CNNs. This is explicitly shown by our experiments where we compared results for different number of distance and magnitude classes. Increasing the number of the magnitude classes, from 2- to 4-class magnitude configuration, we introduce more difficulty upon solving

the magnitude classification problem. From 93%, when we considered only 2 classes the overall accuracy drops to a maximum of 70% for 4 classes during the evaluation of the models (see Figure 5 and the accuracy values associated with the Experiment 2 with respect to the values associated with the Experiments 3, 4, 5, 6, and 8). Moreover, we also notice that for some combination of hyperparameter values, the CNN characterization model is not able to converge for the 4-class magnitude classification (zero accuracy values in Figure 3c). Training on examples with good SNR values does improve the classification slightly. However, we need to have a robust model capable of dealing with low SNR signals, since a continuous signal contains numerous types of event waveforms. For the same reason, we also train our CNN detector using examples of the earthquakes from different distances having magnitude of all ranges. In particular, we treat regional and teleseismic events as signals (positive samples) in the first stage of our CNN pipeline, while in the second stage we specifically design class for them. One could have defined the problem such that all seismic events having epicentral distance <350 km are labeled as positive samples, and earthquakes further away than 350 km are labeled as negative samples in the first stage of our CNN pipeline. Yet, it is less probable that regional and teleseismic events share more common features with noise than with local events, therefore labeling them as noise samples might be harder than separating detection and characterization. When it comes to the distance and the magnitude classification performance and robustness improvement (2-stage of our pipeline), increasing the number of classes is probably not an option. By doing so we would generate more groups of distance and magnitude labels (see groups for this study in Figure S7), that would be nearly empty since there is not enough data. If instead of densely binned classification we switch to regression problem, the challenge could be resolved. Next, we also find that separating a classification problem into two individual tasks does not further improve the overall accuracy, suggesting that one is not introducing any bias by doing either way.

Analyzing the evaluation results of the CNN models in terms of True Negative, True Positive, False Negative, and False Positive, is crucial. In our study, in order to have accurate detections, it was important to set the precision of the model very high, and that implies setting high confidence threshold on the detector probability output. This means that we are losing some new detections for the purpose of not having noise windows classified as signal windows. Naturally, this can be adjusted by setting different confidence threshold on the detector probability output. Furthermore, the level of accuracy we reached with our characterization CNN model is also satisfying. Even though we lose some local events that are being classified as regional events, we are confident that the opposite (having regional and teleseismic events classified as local one) will scarcely occur due to very small probability rate.

One needs to keep in mind that the proposed pipeline does have some limitations. Those limitations relate to the facts that we do not provide the user with the exact event location and we do not know how our pipeline would generalize to other regions. The first point could be addressed by training the CNN characterization model using the back-azimuth values to explore whether this information is also embedded within the raw earthquake waveform. The second point could be addressed by having more systematic approach in which a deep learning model trained on a set of station in a specific region is applied to another region. This might help us understand whether CNN can learn any fault features whatsoever on a distinct set of stations in a distinct region. Nevertheless, we show that our developed CNN pipeline can be successfully applied on the single-station waveforms and still obtain very useful information. Once the models are designed and trained it is possible to extract events very fast and computationally efficiently. All things considered, we developed a pipeline that is particularly efficient when having one station, a sparse catalog and when goal is to reveal the seismicity of a small region.

6. Conclusions

In this study, we have quantified the application of two-stage pipeline consisting of 1-D convolutional neural networks for EQ detection, localization, and characterization. We show that within this framework it is possible to detect near-fault local events of small magnitude using only single-station three-component waveforms and two classification tasks, distance, and magnitude. For these purposes, two CNNs are trained on extensive assortment of EQs from local, regional, and teleseismic events of all magnitudes that defines the robustness of the method. Training and validation experiments performed for extensive hyperparameter grid search, data sets with different SNR values as well as distance and magnitude distribution, different

classes and tasks definition gives us very important insights on what defines a successfully trained CNN model. Furthermore, the general robustness of our pipeline is confirmed within the evaluation process where we are able to reach an accuracy of 97% for event detection (first stage), 94% and 68% for event localization and characterization, respectively (second stage). By applying CNNs on 1-month continuous data, we succeed in detecting all local cataloged events except one and we reveal three times as many new detections as the number of earthquakes in the catalog. By setting a high detection threshold (0.99), all new detections were visually validated as True Positives. Finally, using our two-stage pipeline, we can extract a near-fault catalog of small-magnitude events, and since we also obtain the event time we can even extract the earthquake waveforms and further study, e.g., source parameters, repeating earthquakes. The superiority of CNNs over other EQ detection methods using only one station is their capability to give us a good first estimate of EQ characteristics beside the detection itself, while being very fast and computationally efficient.

Appendix A: Implementation Details

Our two-stage pipeline consisting of 1-D CNNs for earthquake detection, location, and characterization depicted in Figure 2a represent two separated classification problems. In the first stage, we build a detector CNN model that is learning how to classify earthquake windows (positive samples) from noise windows (negative samples). This CNN as an output is giving us a one-element probability vector that tends to zero when the window in question contains only noise and vice versa tends to one when the window in question contain earthquake signal. In the second stage, we build a characterization CNN model that is learning to classify only earthquake windows (positive samples) into two tasks (multitask classification), epicentral distance, and magnitude, where each task is multiclass. As an output, this CNN is producing multiclass-element vector corresponding to the number of tasks and classes. In our case for two tasks, distance, and magnitude, and having two classes for each task means having 1D vector of 4 values (2 tasks \times 2 classes per each task). When CNN is more confident that input sample belongs in a particular distance/magnitude class, then the probability associated with this particular unit of the output vector is closer to one.

Besides the classification problem we want to solve, we also have to define the architecture of the two CNN models. CNN typically consists of input and output layer and several hidden layers. The hidden layers are represented by a series of convolutional layers, where a mathematical operation convolution is applied, that are usually followed by ReLu layers, then additional convolutional, pooling, fully connected, and normalization layers (Albawi et al., 2017; Khan et al., 2020). There are many parameters that needs to be set within the architecture and we use ConvNetQuake model from Perol et al. (2018) as a benchmark. Two architectures that we test in this study are depicted in Figure 2b. These CNNs are 1-D where the input is the 25-s long three-component waveforms samples at 20 Hz, hence the input layer has a dimension 3×500 . Both CNN architectures consist of convolutional layers subsequently followed by ReLu activation function, and finally two fully connected layers. The CNN titled as seven-layer CNN has seven convolutional layers compared to ConvNetQuake, since the input layer in our case is two times smaller than the input in ConvNetQuake. As in ConvNetQuake, the size of the convolutional filter is set to 3. Moreover, we also use 32 channels for layers 1–7 while the input layer having 3 channels and filters are滑ed along the samples by the increment (or stride) equal to 2. Alongside the seven-layered CNN, we also test the performance of the three-layered CNN that has twice as many trainable weights as the seven-layer one. Generating less layers and having shallow architecture is accomplished by changing stride to 2, 4, 3, respectively, for three-convolutional layers. Due to different number of convolutional layers between the seven-layer and three-layer CNN, these two networks are classifying features of different frequency contents (Perol et al., 2018). Hence, there is a general interest to know which model is performing better for EQ-like signals. Since in ConvNetQuake, the problem of detection and localization is unified in one CNN model, to obtain normalized probability distributions we use Sigmoid and Softmax function compared to L_2 -regularized cross-entropy.

As explained in Section 3 to obtain a CNN model there are three stages: training, validation, and evaluation for which we separate our original database in three data sets. In our case, training process is performed by using a minibatch gradient descent, meaning that training data set is split into small batches and for each batch the model error is calculated using a loss function, further the error is back-propagated and the model coefficients are updated using an optimization algorithm. This process is repeated several times or epochs.

Within each epoch subsequently after training we also have a validation process where a model error is calculated for a separate validation data set. We follow how the error (loss function) as well as an accuracy of the model progress for training and validation for each epoch. When the error of the model starts to increase for subsequently 50 epochs we stop training process and keep the model with the minimum error (Yao et al., 2007). Therefore, the training depends on several hyperparameters: the size of the batch used to split the training data (the minibatch size), the optimization algorithm that updated model weights, the learning rate and momentum that control how weight are updated (Ismail Fawaz et al., 2019). For further details see S1 in Supporting Information.

To know whether the model is trained well, as mentioned, during training and validation we monitor the evolution of model's error (loss function) as well as the accuracy (S2 in Supporting Information for the used statistical metric). For the second stage, the loss function is calculated for two tasks (distance and magnitude classification) separately. The final loss is calculated by summing the distance and magnitude loss, since the two function have values of the same order of magnitude. The accuracy for both stages is calculated by setting a confidence threshold to 0.85 on the model probability output to have small rate of False Positive. Finally, the best CNN model, defined by a specific set of parameters, is chosen based on: (a) the minimum reached validation loss value (or the maximum accuracy), (b) the general behavior of the training and validation learning curve. Desired learning curves should show decrease with respect to the epochs, as well as, having training and validation learning curves following each other.

Data Availability Statement

The AquiloNet script to scan continuous seismic traces and obtain raw detections with the distance and the magnitude class is freely available at <https://zenodo.org/record/4709686#.YIFTli0RppQ> and can be cited using the DOI identifier <https://doi.org/10.5281/zenodo.4709686>. Continuous seismic data are downloaded from websites <http://mednet.rm.ingv.it/?more=1>, <http://iside.rm.ingv.it/instruments/network/IV> (INGV Seismological Data Centre, 2006). Most of the computations presented in this study were performed using the GRICAD infrastructure (<https://gricad.univ-grenoble-alpes.fr>), which is supported by Grenoble research communities. Figures 1c and 1d are obtained by the Generic Mapping Tools (Wessel et al., 2019). Some parts of the processing algorithm make use of the SymJAX Python library.

Acknowledgments

We are grateful to two anonymous reviewers for their careful remarks that improved the manuscript. This research received funding from the European Research Council (ERC) under the European Union Horizon 2020 Research and Innovation Program (grant agreements 802777-MONIAUTLS).

References

- Aki, K., & Richards, P. G. (2002). *Quantitative seismology* (2nd ed.). Hardcover: University Science Books.
- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1–6). <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
- Allen, R. (1982). Automatic phase pickers: Their present use and future prospects. *Bulletin of the Seismological Society of America*, 72(6B), S225–S242. <https://doi.org/10.1785/bssa07206b0225>
- Bagh, S., Chiaraluce, L., De Gori, P., Moretti, M., Govoni, A., Chiarabba, C., et al. (2007). Background seismicity in the Central Apennines of Italy: The Abruzzo region case study. *Tectonophysics*, 444(1–4), 80–92. <https://doi.org/10.1016/j.tecto.2007.08.009>
- Beaucé, E., Frank, W. B., & Romanenko, A. (2018). Fast matched filter (FMF): An efficient seismic matched-filter search for both CPU and GPU architectures. *Seismological Research Letters*, 89(1), 165–172. <https://doi.org/10.1785/0220170181>
- Bergen, K. J., & Beroza, G. C. (2018). Detecting earthquakes over a seismic network using single-station similarity measures. *Geophysical Journal International*, 213(3), 1984–1998. <https://doi.org/10.1093/gji/ggy100>
- Bergen, K. J., Johnson, P. A., de Hoop, M. V., & Beroza, G. C. (2019). Machine learning for data-driven discovery in solid earth geoscience. *Science*, 363(6433), eaau0323. <https://doi.org/10.1126/science.aau0323>
- Bouchon, M., Durand, V., Marsan, D., Karabulut, H., & Schmittbuhl, J. (2013). The long precursory phase of most large interplate earthquakes. *Nature Geoscience*, 6(4), 299–302. <https://doi.org/10.1038/ngeo1770>
- Bouchon, M., Karabulut, H., Aktar, M., Ozalaybey, S., Schmittbuhl, J., & Bouin, M.-P. (2011). Extended nucleation of the 1999 Mw 7.6 Izmit earthquake. *Science*, 331(6019), 877–880. <https://doi.org/10.1126/science.1197341>
- Brodsky, E. E. (2019). The importance of studying small earthquakes. *Science*, 364(6442), 736–737. <https://doi.org/10.1126/science.aax2490>
- Castello, B., Selvaggi, G., Chiarabba, C., & Amato, A. (2006). *Csi catalogo della sismicità italiana 1981–2002*. Rome: Istituto Nazionale di Geofisica e Vulcanologia/Centro Nazionale Terremoti. Retrieved from <http://www.ingv.it/CSI/>
- Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *Proceedings of the British Machine Vision Conference*. BMVA Press. <https://doi.org/10.5244/C.28.6>
- Chiarabba, C., Jovane, L., & DiStefano, R. (2005). A new view of Italian seismicity using 20 years of instrumental recordings. *Tectonophysics*, 395(3–4), 251–268. <https://doi.org/10.1016/j.tecto.2004.09.013>
- Chiaraluce, L., Valoroso, L., Anselmi, M., Bagh, S., & Chiarabba, C. (2009). A decade of passive seismic monitoring experiments with local networks in four Italian regions. *Tectonophysics*, 476(1–2), 85–98. <https://doi.org/10.1016/j.tecto.2009.02.013>
- Cui, Z., Chen, W., & Chen, Y. (2016). *Multi-scale convolutional neural networks for time series classification*. arXiv:abs/1603.06995.
- Dai, H., & MacBeth, C. (1995). Automatic picking of seismic arrivals in local earthquake data using an artificial neural network. *Geophysical Journal International*, 120(3), 758–774. <https://doi.org/10.1111/j.1365-246X.1995.tb01851.x>

- de Arcangelis, L., Godano, C., Grasso, J. R., & Lippiello, E. (2016). Statistical physics approach to earthquake occurrence and forecasting. *Physics Reports*, 628, 1–91. <https://doi.org/10.1016/j.physrep.2016.03.002>
- De Luca, G., Cattaneo, M., Monachesi, G., & Amato, A. (2009). Seismicity in Central and Northern Apennines integrating the Italian national and regional networks. *Tectonophysics*, 476(1–2), 121–135. <https://doi.org/10.1016/j.tecto.2008.11.032>
- Ellsworth, W. L. (2019). *From foreshocks to mainshocks: Mechanisms and implications for Earthquake nucleation and rupture propagation* (Vol. 202). IOS Press. <https://doi.org/10.3254/978-1-61499-979-9-95>
- Ellsworth, W. L., & Bulut, F. (2018). Nucleation of the 1999 Izmit earthquake by a triggered cascade of foreshocks. *Nature Geoscience*, 11(7), 531–535. <https://doi.org/10.1038/s41561-018-0145-1>
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8), 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- Gibbons, S. J., & Ringdal, F. (2006). The detection of low magnitude seismic events using array-based waveform correlation. *Geophysical Journal International*, 165(1), 149–166. <https://doi.org/10.1111/j.1365-246X.2006.02865.x>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (Vol. 27, pp. 2672–2680). Curran Associates, Inc.
- Gutenberg, B., & Richter, C. F. (1955). Magnitude and energy of earthquakes. *Nature*, 176(4486), 795. <https://doi.org/10.1038/176795a0>
- INGV Seismological Data Centre. (2006). *Rete Sismica Nazionale* (RSN). Italy: Istituto Nazionale di Geofisica e Vulcanologia (INGV). <https://doi.org/10.13127/SD/X0FXNH7QFY>
- Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: A review. *Data Mining and Knowledge Discovery*, 33(4), 917–963. <https://doi.org/10.1007/s10618-019-00619-1>
- Jones, L. M., & Molnar, P. (1976). Frequency of foreshocks. *Nature*, 262(5570), 677–679. <https://doi.org/10.1038/262677a0>
- Jones, L. M., & Molnar, P. (1979). Some characteristics of foreshocks and their possible relationship to earthquake prediction and premonitory slip on faults. *Journal of Geophysical Research*, 84(B7), 3596–3608. <https://doi.org/10.1029/JB084iB07p03596>
- Kanamori, H. (1981). The nature of seismicity patterns before large earthquakes. In *Earthquake prediction* (pp. 1–19). American Geophysical Union (AGU). <https://doi.org/10.1029/ME004p0001>
- Kato, A., Obara, K., Igarashi, T., Tsuruoka, H., Nakagawa, S., & Hirata, N. (2012). Propagation of slow slip leading up to the 2011 Mw 9.0 Tohoku-Oki earthquake. *Science*, 335(6069), 705–708. <https://doi.org/10.1126/science.1215141>
- Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455–5516. <https://doi.org/10.1007/s10462-020-09825-6>
- Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2019). Machine learning in seismology: Turning data into insights. *Seismological Research Letters*, 90(1), 3–14. <https://doi.org/10.1785/0220180259>
- Kriegerowski, M., Petersen, G. M., Vasyura-Bathke, H., & Ohrnberger, M. (2018). A Deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms. *Seismological Research Letters*, 90(2A), 510–516. <https://doi.org/10.1785/0220180320>
- Küperkoch, L., Meier, T., Lee, J., Friederich, W., & Group, E. W. (2010). Automated determination of P-phase arrival times at regional and local distances using higher order statistics. *Geophysical Journal International*, 181(2), 1159–1170. <https://doi.org/10.1111/j.1365-246X.2010.04570.x>
- Le Guennec, A., Malinowski, S., & Tavenard, R. (2016). Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, Riva Del Garda, Italy. Retrieved from <https://halshs.archives-ouvertes.fr/halshs-01357973>
- Li, Z., Meier, M.-A., Hauksson, E., Zhan, Z., & Andrews, J. (2018). Machine learning seismic wave discrimination: Application to earthquake early warning. *Geophysical Research Letters*, 45, 4773–4779. <https://doi.org/10.1029/2018GL077870>
- Lomax, A., Michelini, A., & Jozinović, D. (2019). An investigation of rapid earthquake characterization using single-station waveforms and a convolutional neural network. *Seismological Research Letters*, 90(2A), 517–529. <https://doi.org/10.1785/0220180311>
- Lubbers, N., Bolton, D. C., Mohd-Yusof, J., Marone, C., Barros, K., & Johnson, P. A. (2018). Earthquake catalog-based machine learning identification of laboratory fault states and the effects of magnitude of completeness. *Geophysical Research Letters*, 45, 13269–13276. <https://doi.org/10.1029/2018GL079712>
- Magrini, F., Jozinović, D., Cammarano, F., Michelini, A., & Boschi, L. (2020). Local earthquakes detection: A benchmark dataset of 3-component seismograms built on a global scale. *Artificial Intelligence in Geosciences*, 1, 1–10. <https://doi.org/10.1016/j.aiagi.2020.04.001>
- Mignan, A. (2011). Retrospective on the accelerating seismic release (ASR) hypothesis: Controversy and new horizons. *Tectonophysics*, 505(1), 1–16. <https://doi.org/10.1016/j.tecto.2011.03.010>
- Mignan, A., & Broccardo, M. (2020). Neural network applications in earthquake prediction (1994–2019): Meta-analytic and statistical insights on their limitations. *Seismological Research Letters*, 91(4), 2330–2342. <https://doi.org/10.1785/0220200021>
- Mousavi, S. M., & Beroza, G. C. (2020a). Bayesian-deep-learning estimation of earthquake location from single-station observations. *IEEE Transactions on Geoscience and Remote Sensing*, 58(11), 8211–8224. <https://doi.org/10.1109/tgrs.2020.2988770>
- Mousavi, S. M., & Beroza, G. C. (2020b). A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, 47, e2019GL085976. <https://doi.org/10.1029/2019GL085976>
- Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020). Earthquake transformer—An attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, 11(1), 3952. <https://doi.org/10.1038/s41467-020-17591-w>
- Mousavi, S. M., Sheng, Y., Zhu, W., & Beroza, G. C. (2019). Stanford earthquake dataset (stead): A global data set of seismic signals for AI. *IEEE Access*, 7, 179464–179476. <https://doi.org/10.1109/ACCESS.2019.2947848>
- Mousavi, S. M., Zhu, W., Sheng, Y., & Beroza, G. C. (2019). CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection. *Scientific Reports*, 9(1), 10267. <https://doi.org/10.1038/s41598-019-45748-1>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., & Chintala, S. (2019). *PyTorch: An imperative style, high-performance deep learning library*. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32, pp. 8024–8035). Curran Associates, Inc. Retrieved from <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- Peng, Z., & Zhao, P. (2009). Migration of early aftershocks following the 2004 Parkfield earthquake. *Nature Geoscience*, 2(12), 877–881. <https://doi.org/10.1038/ngeo697>
- Perol, T., Gharbi, M., & Denolle, M. (2018). Convolutional neural network for earthquake detection and location. *Science Advances*, 4(2), e1700578. <https://doi.org/10.1126/sciadv.1700578>
- Rodríguez-Pradilla, G., & Eaton, D. W. (2019). The application of coda and energy methods for magnitude estimation of microseismic events. *Seismological Research Letters*, 90(3), 1296–1307. <https://doi.org/10.1785/0220180366>

- Ross, Z. E., Rollins, C., Cochran, E. S., Hauksson, E., Avouac, J.-P., & Ben-Zion, Y. (2017). Aftershocks driven by after slip and fluid pressure sweeping through a fault-fracture mesh. *Geophysical Research Letters*, 44, 8260–8267. <https://doi.org/10.1029/2017GL074634>
- Ross, Z. E., Yue, Y., Meier, M.-A., Hauksson, E., & Heaton, T. H. (2019). Phaselink: A deep learning approach to seismic phase association. *Journal of Geophysical Research: Solid Earth*, 124, 856–869. <https://doi.org/10.1029/2018JB016674>
- Sánchez-Reyes, H., Essing, D., Beaucé, E., & Poli, P. (2020). The imbricated foreshock and aftershock activities of the Balsorano (Italy) Mw 4.4 normal fault earthquake and implications for earthquake initiation. *Seismological Research Letters*, 92(3), 1926–1936. <https://doi.org/10.1785/0220200253>
- Saragiotis, C. D., Hadjileontiadis, L. J., & Panas, S. M. (2002). PAI-S/K: A robust automatic seismic p phase arrival identification scheme. *IEEE Transactions on Geoscience and Remote Sensing*, 40(6), 1395–1404. <https://doi.org/10.1109/TGRS.2002.800438>
- Seydoux, L., Balestrieri, R., Poli, P., Hoop, M. D., Campillo, M., & Baraniuk, R. (2020). Clustering earthquake signals and background noises in continuous seismic data with unsupervised deep learning. *Nature Communications*, 11(1), 3972. <https://doi.org/10.1038/s41467-020-17841-x>
- Shelly, D. R., Beroza, G. C., & Ide, S. (2007). Non-volcanic tremor and low-frequency earthquake swarms. *Nature*, 446(7133), 305–307. <https://doi.org/10.1038/nature05666>
- Shi, P., Seydoux, L., & Poli, P. (2020). Unsupervised learning of seismic wavefield features: Clustering continuous array seismic data during the 2009 L'Aquila earthquake. *Journal of Geophysical Research: Solid Earth*, 126, e2020JB020506. <https://doi.org/10.1029/2020JB020506>
- Socquet, A., Valdes, J. P., Jara, J., Cotton, F., Walpersdorf, A., Cotte, N., et al. (2017). An 8 month slow slip event triggers progressive nucleation of the 2014 Chile megathrust. *Geophysical Research Letters*, 44, 4046–4053. <https://doi.org/10.1002/2017GL073023>
- Sugan, M., Kato, A., Miyake, H., Nakagawa, S., & Vuan, A. (2014). The preparatory phase of the 2009 Mw 6.3 L'Aquila earthquake by improving the detection capability of low-magnitude foreshocks. *Geophysical Research Letters*, 41, 6137–6144. <https://doi.org/10.1002/2014GL061199>
- Tertulliani, A., Rossi, A., Cucci, L., & Vecchi, M. (2009). L'Aquila (Central Italy) earthquakes: The predecessors of the April 6, 2009 event. *Seismological Research Letters*, 80(6), 1008–1013. <https://doi.org/10.1785/gssr.80.6.1008>
- U.S. Geological Survey. (2017). *Earthquake facts and statistics*. Retrieved from <https://earthquake.usgs.gov/earthquakes/browse/stats.php>
- Valoroso, L., Chiaraluce, L., Piccinini, D., Di Stefano, R., Schaff, D., & Waldhauser, F. (2013). Radiography of a normal fault system by 64,000 high-precision earthquake locations: The 2009 L'Aquila (central Italy) case study. *Journal of Geophysical Research: Solid Earth*, 118, 1156–1176. <https://doi.org/10.1002/jgrb.50130>
- van den Ende, M. P. A., & Ampuero, J.-P. (2020). Automated seismic source characterization using deep graph neural networks. *Geophysical Research Letters*, 47, e2020GL088690. <https://doi.org/10.1029/2020GL088690>
- Vuan, A., Sugan, M., Amati, G., & Kato, A. (2018). Improving the detection of low-magnitude seismicity preceding the Mw 6.3 L'Aquila earthquake: Development of a scalable code based on the cross correlation of template earthquakes. *Bulletin of the Seismological Society of America*, 108(1), 471–480. <https://doi.org/10.1785/0120170106>
- Wessel, P., Luis, J. F., Uieda, L., Scharroo, R., Wobbe, F., Smith, W. H. F., & Tian, D. (2019). The generic mapping tools version 6. *Geochemistry, Geophysics, Geosystems*, 20, 5556–5564. <https://doi.org/10.1029/2019GC008515>
- Wu, Y., Lin, Y., Zhou, Z., Bolton, D. C., Liu, J., & Johnson, P. (2019). DeepDetect: A cascaded region-based densely connected network for seismic event detection. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1), 62–75. <https://doi.org/10.1109/TGRS.2018.2852302>
- Wu, Y., Lin, Y., Zhou, Z., & Delorey, A. (2018). Seismic-Net: A deep densely connected neural network to detect seismic events. arXiv:1802.02241 [cs, eess].
- Yao, Y., Rosasco, L., & Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, 26(2), 289–315. <https://doi.org/10.1007/s00365-006-0663-2>
- Yoon, C. E., O'Reilly, O., Bergen, K. J., & Beroza, G. C. (2015). Earthquake detection through computationally efficient similarity search. *Science Advances*, 1(11), e1501057. <https://doi.org/10.1126/sciadv.1501057>
- Zhu, W., & Beroza, G. C. (2018). PhaseNet: A deep-neural-network-based seismic arrival time picking method. *Geophysical Journal International*, 216(1), 261–273. <https://doi.org/10.1093/gji/ggy423>
- Zhu, W., Mousavi, S. M., & Beroza, G. C. (2019). Seismic signal denoising and decomposition using deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11), 9476–9488. <https://doi.org/10.1109/TGRS.2019.2926772>
- Zhu, W., Mousavi, S. M., & Beroza, G. C. (2020). Chapter four: Seismic signal augmentation to improve generalization of deep neural networks. In *Machine learning in geosciences*. Elsevier.