Seismic Event Detection via Deep Multi-Task Learning

Yang Yu†, Lei Zhang‡, Jiakai Shen*, Qingcai Wang† and Guiquan Liu†
† *School of Computer Science and Technology, University of Science and Technology of China, Hefei, China*
Email:{koinu, wqc1024}@mail.ustc.edu.cn, gqliu@ustc.edu.cn (*corresponding author*)
‡*Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education,*
School of Computer Science and Technology, Anhui University, Hefei 230039, China
Email: zl@ahu.edu.cn (*corresponding author*)
**Department of Computer Science and Technology, Nanjing University, Nanjing, China*
Email: mg1633063@smail.nju.edu.cn

***Abstract*—It is fundamental to detect seismic events reliably and efficiently when processing continuous waveform data recorded by seismic stations. Recently, convolutional neural network (CNN) based detecting methods have been proposed for seismic events detection and obtained great success in this area, where the learning of seismic event detecting network of all seismic stations is considered as one learning task and numerous labeled data need to be collected for training the detecting network. However, they tend to ignore the differences between seismic stations caused by geographic position. Moreover, due to a few seismic activities and high cost of manual data labeling, in some areas, the labeled data for seismic event detecting tasks is insufficient. Under this condition, these methods always encounter over-fitting problem leading to bad detection performance. In this paper, we propose a multi-task based framework based on convolutional neural network for accurate seismic event detection under the condition of insufficient labeled data. Specifically, we first cluster the seismic stations into several station clusters and treat the learning of seismic event detecting network of every station cluster as a learning task, and then we propose a deep multi-task network named *detectMTL4* among multiple tasks. Experimental results on a real-world seismic dataset with nine stations demonstrate the effectiveness of the proposed framework, especially when the labeled data is insufficient.**

***Index Terms*—Seismic event detection, Multi-task learning, Deep learning.**

I. INTRODUCTION

Professional seismic stations are widely distributed, which record continuous waveform data day and day. The continuous waveform record usually consists of three components, two of which record waveform in the horizontal direction, the rest one records waveform in the vertical direction. Fig. 1 gives an illustration of a three-component seismic event record. Based on these collected continuous waveform records, one important task for analysis is to effectively detect seismic event.

The typical way to detect seismic event is to slice waveform segment from continuous record and adopt some method to determine whether this segment contains a seismic event or not. In the early work, seismic events were manually detected. However, manual detection cannot meet the demand of real-time processing system. Therefore, many automatic detecting methods were proposed [1]–[3]. These methods detect seismic events from a signal processing perspective. The detection happens when the extracted waveform attributes exceed the predetermined or dynamically specified threshold value. However, these methods suffer from three challenges: first, these detecting methods are sensitive to noise. The false positive rate and false negative rate of these methods will rise when the waveform records are of low signal-to-noise ratio (SNR). Secondly, these methods need to design waveform attributes which require a good command of seismological knowledge and may fail to fully utilize the information contained in the seismic event records. Thirdly, these methods will not perform well without a lot of effort on manual parameter tuning.

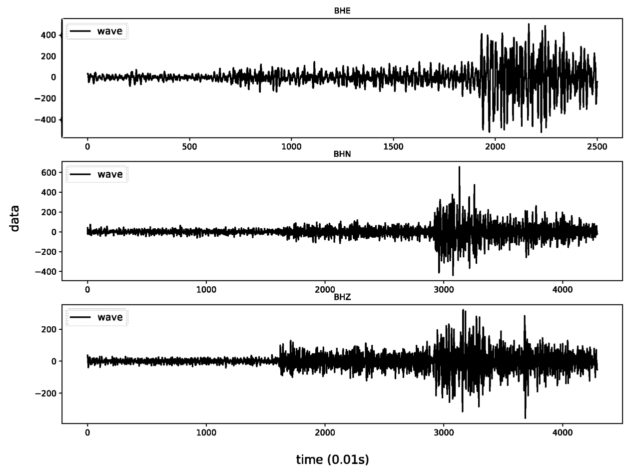

Fig. 1: A seismic event record, where BHE, BHN and BHZ represent the three components respectively.

With the development of deep learning, convolutional neural network (CNN) based detecting methods were proposed [4], [5]. However, these methods face two main challenges: firstly, these methods tend to ignore the fact that seismic stations are usually widely distributed. Fig. 2 shows the locations of seismic stations around Northern California area in USA. From this figure, it can be found that there are nine seismic stations which are widely distributed. When an earthquake happens, due to distinct geographic position of seismic stations, the waveform received by seismic stations may differ. Secondly, these methods train network with a large number of labeled data. However, in some areas, due to a few seismic activities and high cost of manual data labeling, there are insufficient labeled data. Under this condition, the CNN-based detecting methods tend to perform poorly due to over-fitting problem.

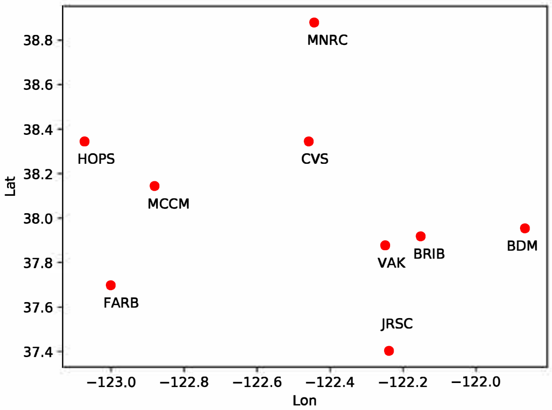

Fig. 2: The location of nine seismic stations in Northern California area of USA.

To solve the above challenges, in this paper, we propose a deep multi-task learning based framework to detect seismic event with a few labeled data. Considered that the difference between seismic stations, it is impractical to regard learning detecting network of a seismic station as a task, since the widely distributed seismic stations bring difficulties in determining the relationships between tasks. Therefore, in this paper, we first cluster seismic stations into several station clusters and regard learning detecting network of a station cluster as a task. Then, we use the proposed deep multi-task model to obtain the detecting network of every station cluster. Specifically, the detecting network consists of two components: shared layers among tasks and task-specific layers. With the use of multi-task learning, the over-fitting problem can be alleviated and the performance of CNN-based detecting methods with few labeled data can be improved. In summary, we made several contributions as follows:

- With full consideration of difference among seismic stations, we do not regard learning detecting network of all stations as a task. We perform clustering on seismic stations based on geographic positions to get several station clusters and regard learning detecting network of a station cluster as a task.
- In order to alleviate over-fitting problem, we propose a deep multi-task model to obtain detecting network of every station cluster. The detecting network consists of two parts: the shared layers which are used for sharing common knowledge and the task-specific layers which are used for making up the difference between tasks.
- The experimental results on a real-world seismic dataset with nine stations demonstrate the promising performance of the proposed framework for seismic event detection, which also indicates that our framework can greatly improve the detection performance especially while the labeled data is insufficient.

The rest of paper is organized as follows. We review the related studies in Section 2. The proposed framework is introduced in Section 3. Experimental results are described in Section 4. Finally, we conclude our work and give the future work in Section 5.

II. Related Works

The related studies can be grouped into two categories: seismic event detecting methods and multi-task learning methods.

A. *Existing Seismic Event Detecting Methods*

The most popular seismic event detecting method is the short term average to long term average ratio (STA/LTA) [1]. The principle of STA/LTA is readily comprehensible. The detection happens when the ratio between a short term average of a characteristic function and a long term average of the same characteristic function exceeds the threshold value. The STA/LTA of point i is computed as follows:

$$Ratio_i = \frac{\frac{1}{ns}\sum_{j=i-ns+1}^{i} CF(j)}{\frac{1}{nl}\sum_{k=i-nl+1}^{i} CF(k)} \tag{1}$$

where ns denotes the length of short term, nl denotes the length of long term and CF denotes the characteristic function.

Based on different computation approaches, STA/LTA can be sorted in two categories: classical STA/LTA (CSL) and recursive STA/LTA (RSL) [2]. The CSL and RSL of point i are computed as Eq. (2) and Eq. (3) respectively.

$$Ratio_i = \frac{STA_{i-1} + \frac{CF(i)-CF(i-ns)}{ns}}{LTA_{i-1} + \frac{CF(i-ns-1)-CF(i-ns-nl-1)}{nl}} \tag{2}$$

$$Ratio_i = \frac{STA_{i-1} + \frac{CF(i)-STA_{i-1}}{ns}}{LTA_{i-1} + \frac{CF(i-ns-1)-LTA_{i-1}}{nl}} \tag{3}$$

STA/LTA and its variants are simple and detect seismic events at high speed. However, they tend to perform poorly when processing waveform records of low signal-to-noise ratio (SNR). Meanwhile, these methods need a lot of effort on parameter tuning, such as the selection of threshold and characteristic function [3].

Convolutional Neural Network (CNN) [6] is a classical kind of neural network specialized for feature extraction, which has been widely used in various fields, such as video surveillance, mobile robot vision and seismic prediction [7]–[9]. The main parts of CNNs are convolutional layer and pooling layer. Convolutional layer utilizes filters to extract features whose characteristics are sparse interactions, parameter sharing and equivariant representations. Pooling layer maintains useful information while reducing the volume of data based on sub-sampling theory. High level features can be obtained by stacking layers of CNNs. With the rapid development of deep learning and success of CNN in object detection, various CNN-based seismic event detecting methods were proposed. For example, in [4], a highly scalable convolutional neural network was proposed for seismic event detection and location.

In [5], a cascaded region-based convolutional neural network was proposed for seismic event detection. These CNN-based seismic event detecting methods have demonstrated their effectiveness in seismic event detection, however, they tend to ignore the difference between seismic stations and regard learning detecting network of all stations as a task. With sufficient labeled data, these methods perform well on seismic event detection. But the labeled data for seismic event detecting tasks in real collected data is usually insufficient. Under this condition, the performance of existing CNN-based methods will degrade. In this paper, with consideration of difference between seismic stations, we propose a deep multi-task learning framework to improve the prediction performance on seismic event detection under insufficient labeled data.

B. Multi-Task Learning

Usually, in order to obtain a good learner, we need a large number of labeled data. However, due to the nature of problem itself, high cost of data labeling and other reasons, sometimes it is hard to collect enough labeled data. For data insufficient problem, Multi-Task Learning (MTL) [10] is a good solution. MTL is inspired by human being learning ability, aiming at how to improve the generalization performance of multiple related tasks with limited labeled data by leverage the common knowledge among them.

The key challenge in MTL is to exploit relationships between tasks. At early stage, prior information was imposed on task relationships. In [11], model parameters of all tasks were assumed to be close to each other, a regularizer was proposed to enforce the model parameters of all tasks to be close to the mean one. However, such prior information is difficult to obtain. In recent years, advanced MTL methods were proposed to learn task relationships. For example, multi-task feature learning approaches were proposed, which can be further sorted into feature selection and feature transformation approaches. These approaches aims at learning a common feature representation from original feature space with or without transformation [12], [13]. Multi-task low-rank approaches assume that models share the same low rank subspace [14]. In practice, it is too restrictive to constrain all the tasks to share the same structure. In [15], it was assumed that the model of task consists of a shared low-dimensional subspace and a task-specific component. In [16], it was assumed that the model of task consists of a group sparse component and a task-specific sparse component. Different from traditional MTL, multi-task task clustering approaches assume that the tasks can be partitioned into several clusters, where tasks within a cluster are related [17]. There are different methods to detect cluster structure, such as Dirichlet process [18], integer programming [19], identifying representative tasks [20] and so on.

In deep learning, MTL can be classified into two categories: hard parameter sharing where the model of every task consists of shared layers among tasks and task-specific layers and soft parameter sharing where each task has its own model and the distance between parameters of the model is regularized to enforce parameters to be similar [21]. MTL has many applications in real-world tasks, such as natural language processing [22], [23], computer vision [24], [25] and so on. MTL has also been incorporated with other disciplines, such as multi-task multi-view learning [26], multi-task reinforcement learning [27], multi-task multi-label learning [28] and so on. In this paper, the technique of MTL is applied on seismic event detecting. Unlike typical way of using MTL, we do not regard learning detecting network of every station as a task, due to the fact that seismic stations are widely distributed and it is hard to determine the relationships between tasks. To this end, we first perform seismic station clustering and then regard the detecting network of each station cluster as a task.

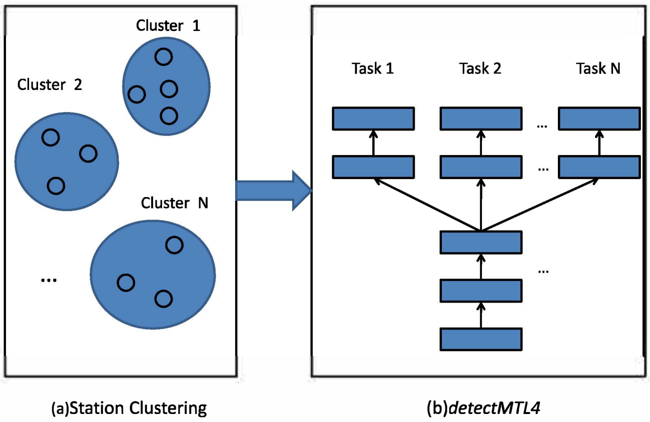

Fig. 3: The procedure of the proposed framework.

III. Proposed Framework

As shown in Fig. 3, the proposed framework consists of two steps. In the first step, we cluster seismic stations based on geographic position to obtain several station clusters and regard learning detecting network of each station cluster as a task. In the second step, we use the proposed deep multi-task model named $detectMTL4$ to obtain the detecting network of every task where the network consists of shared layers and task-specific layers.

A. Station Clustering

Existing CNN-based detecting methods tend to ignore the difference between seismic stations and regard learning the detecting network of all stations as a task. However, seismic stations are widely and uniformly distributed. When an earthquake happens, due to different geographic positions, the waveform record collected by seismic stations may differ from each other. When there are a large number of labeled data, the performance of existing CNN-based detecting methods will not be affected a lot by the ignorance of differences between seismic stations. However, when the volume of training data decreases, the generalization performance of these methods will degrade a lot.

To address the above problem, multi-task learning is a good solution. However, it is impractical to directly treat learning detecting network of every seismic station as a task. Some stations situate closely and some stations situate far away, if learning the detecting network of every seismic station is regarded as a task, it is hard to measure the relatedness between tasks. Therefore, in this paper, we first cluster seismic stations to obtain several station clusters. It is intuitive that the closer seismic stations situate, the more similar the waveform records that they collected. Therefore, learning the detecting network of seismic stations of a station cluster is considered as a task. In this paper, we adopt a popular and simple K-means clustering algorithm to cluster the seismic stations.

B. Proposed Network

In this paper, we apply the basic CNN as the basic detecting network of every task and utilize hard parameter sharing deep multi-task learning to form the proposed *detectMTL4*, where the network structure is shown in Fig. 4.

To be specific, the detecting network of each task consists of two parts: shared layers and task-specific layers. The shared layers are used for sharing common knowledge while the task-specific layers are used for making up differences between tasks. The structure of detecting network of every task stays the same, while the difference lies on the parameters of task-specific layers. The shared layers start from $conv1$ layer to $pooling4$ layer, which provide the same feature representations for each task. The task-specific layers make up the difference between tasks and learn task-specific feature representations based on shared feature representations, which starts from $conv5$ layer to the end.

The detailed information about *detectMTL4* is illustrated as follows. The convolutional layers we adopt are 1-dimensional, the length of feature map is 6 and the stride is 1. $conv1$ convolutional layer has 8 feature maps, $conv2$ convolutional layer has 16 features and the rest of convolutional layers has 32 feature maps. The pooling layers we adopt are 1-dimension, the pooling strategy is max-pooling and the window size of pooling layers is 3. The stride of $pooling1$ pooling layer and $pooling2$ pooling layer is 2, the stride of the remaining pooling layers is 4. The padding strategy of convolutional layers and pooling layers is $SAME$ padding. Since detecting seismic event task is a binary classification problem, the loss function we adopted is cross-entropy loss function.

IV. Experiments

In this section, we first give experimental settings, including datasets, comparison algorithms, the evaluation criterions, and parameters. Then, we present the experimental results and analysis.

A. Experimental Settings

1) Dataset: In this paper, we evaluate the proposed framework on a real-world dataset named *Napa* [29]. The *Napa* dataset contains continuous waveform records in 2014, before and after Napa earthquake in North California. There are 1,360 events recorded by nine stations. Table I gives the characteristic of *Napa* dataset sorted by seismic station. For *Napa* dataset, the length of segment of seismic event we sliced is 40 seconds. To obtain negative samples, we slice segment

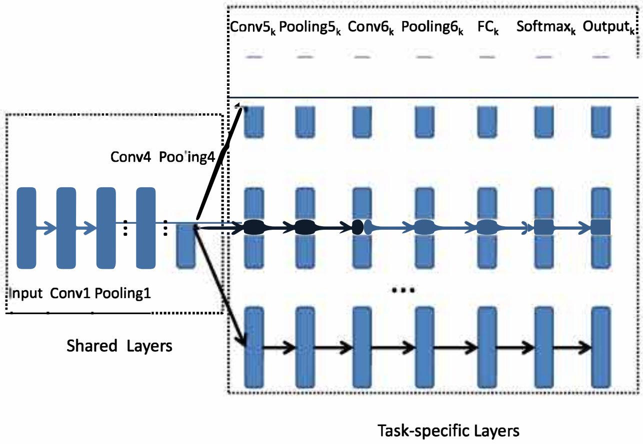

Fig. 4: The network structure of the proposed *detectMTL4*.

TABLE I: Characteristics of seismic stations in *Napa* dataset.

Station	Longitude	Latitude	Number of events
BDM	-121.86554	37.95397	154
$BRIB$	-122.15179	37.91886	214
CVS	-122.45840	38.34526	316
$FARB$	-123.00110	37.69782	53
$HOPS$	-123.07234	38.99349	28
$JRSC$	-122.23868	37.40373	61
$MCCM$	-122.88018	38.14478	227
$MNRC$	-122.44277	38.87874	79
VAK	-122.24889	37.87753	228

of waveform record before Napa earthquake with the length of 40 seconds, the number of negative samples is same with the number of seismic events of each seismic station. All the data are preprocessed with normalization, detrending and Bandpass filter [30]. 10% of *Napa* dataset has been divided in advance as validation set.

2) Comparison schemes and evaluation criterions: In our experiments, we evaluate the proposed framework with four competing methods. The first one is the single task learning method which regards learning the detecting network of all seismic stations as a task and the structure of detecting network is the same with *detectMTL4* with all layers shared, denoted as *SingleTask*. The second one is *ConvNetQuake* [4], which is a representative method of existing CNN-based detecting methods. The other two methods are traditional detecting methods, CSL [2] and RSL [2].

Since seismic event detection is a classification problem, $F1$ score and *Accuracy* are employed to evaluate the experimental results.

3) Experimental parameters: For fair comparisons, the recommended parameters values are adopted for all the comparison algorithms, which were suggested in their original papers. For the proposed framework, we add L2 regularizer on weights of $conv1$ and $conv2$ layers and use dropout and early stopping tricks to prevent over-fitting, the dropout rate is set to be 0.5. The learning algorithm we adopted is Adam [31]. The learning rate is set to 0.0005. The way we train the detecting network of each task is alternative training, which is shown in Fig. 5. As can be observed from this figure, each task is trained alternatively and the batch size of each task is set according to the proportion of seismic events in each station cluster so that we can keep the statistical characteristic of *Napa* dataset. For convolutional layers and fully connected layers, we take ReLU as activation function and initialize weights with truncated normal distribution with standard deviation of 0.1. For Softmax layer, w is initialized with truncated normal distribution, with standard deviation of 0.05, b is initialized all zeroes. The k in K-means clustering is set to be 3.

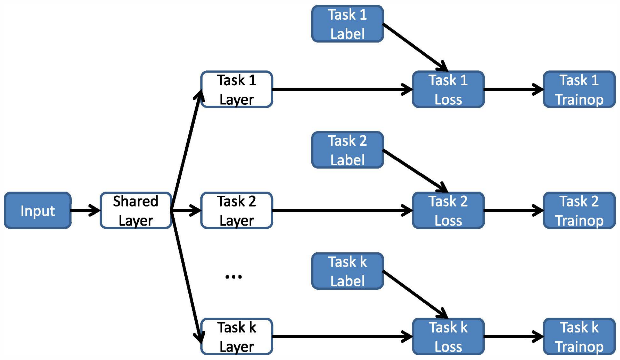

Fig. 5: The computational graph of alternative training.

B. Experimental Results

1) Effectiveness of proposed framework: We divide the adopted data with nine stations into three station clusters. Specifically, Station cluster-1 includes $FARB$, $HOPS$ and $MCCM$, station cluster-2 includes BDM, $BRIB$, $JRSC$ and VAK, station cluster-3 includes CVS and $MNRC$.

Fig. 6 shows the performance of comparison methods in terms of $F1$ score and *Accuracy* under different training ratio. It can be found that under different training ratio, the proposed *detectMTL4* performs better than *SingleTask*, which verifies the effectiveness of *detectMTL4*. It is mainly because in *detectMTL4*, besides the shared layers transferring common knowledge, there are task-specific layers that keep the instinct characteristic of each task. In fact, *SingleTask* can be regarded as a special situation of *detectMTL4* where all the seismic stations are treated within a cluster. Also, *detectMTL4* performs better than *ConvNetQuake*. It is mainly because the performance of *ConvNetQuake* degrade due to the over-fitting problem. Moreover, we can find that *detectMTL4*, *SingleTask* and *ConvNetQuake* perform better than CSL and RSL, which also demonstrates the effectiveness of CNN-based detecting methods. Last but not least, it can be also found that the performance gap between the proposed *detectMTL4* and CNN-based detecting methods (*SingleTask* and *ConvNetQuake*) becomes large when the training ratio decreases. In other words, the proposed method can greatly improve the performance of CNN-based detecting methods especially when the labeled data is insufficient.

2) Parameter sensitivity analysis: There is one important parameter in the proposed model, that is, the number of shared layers. To this end, we analyze the sensitivity of the number of shared layers under the training ratio of 80%. Table II gives the details about models with different number of shared layers. Table III presents the performance of different models shown in Table II in terms of $F1$ score and *Accuracy*. From this table, we can see that the performance of *detectMTL4* first increases with the number of shared layers increases, and get the best performance when the number of shared layers increases reaches four. Then, the performance decreases with the number of shared layers increases. This is because that when increasing the shared layers, the model will degrade to *SingleTask* where all the seismic stations are treated within a cluster. In addition, when decreasing the shared layers, the model will degrade to another kind of single task learning where learning detecting network of each station cluster is regarded as an independent task with no shared layers, under this condition, the available labeled data for each task with be further reduced and the over-fitting problem will be further serious. Thus, in this paper, the number of shared layers is suggested as four.

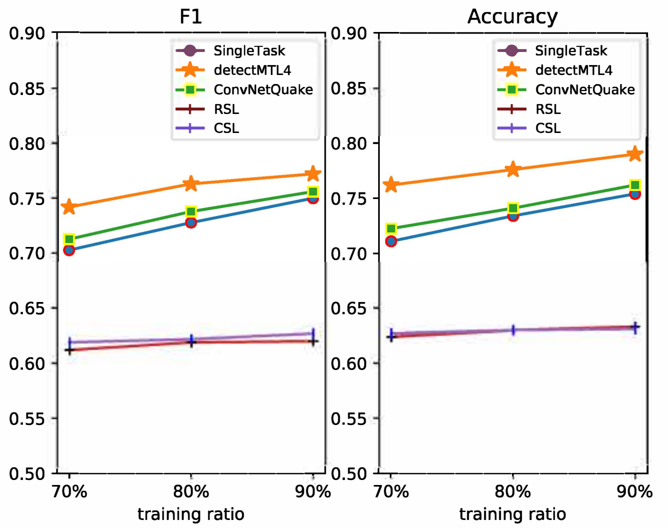

Fig. 6: The comparison results of four baselines and our method on *Napa* dataset in terms of $F1$ score and *Accuracy*.

TABLE II: Models with different number of shared layers.

Model	Shared layers
detectMTL2	from $conv1$ to $pooling2$
detectMTL3	from $conv1$ to $pooling3$
detectMTL4	from $conv1$ to $pooling4$
detectMTL5	from $conv1$ to $pooling5$
detectMTL6	from $conv1$ to $pooling6$

TABLE III: The comparison results of models with different shared layers in terms of $F1$ score and *Accuracy*.

Model	F1 score	Accuracy
detectMTL2	0.738	0.759
detectMTL3	0.757	0.762
detectMTL4	**0.764**	**0.776**
detectMTL5	0.759	0.771
detectMTL6	0.742	0.764

V. Conclusion and Future Work

In this paper, we studied the problem of seismic event detection with a few labeled data. Unlike most existing CNN-based detecting methods, we proposed a deep multi-task based framework to detect seismic events. To be specific, the proposed framework first clustered seismic stations based on geographic positions and regard learning the detecting network of each station cluster as one task, then utilized the proposed *detectMTL4* to obtain the detecting network of each task. Specifically, *detectMTL4* consists of shared layers that were used for sharing common knowledge between tasks and task-specific layers that were used for making up for the difference between seismic stations. Finally, we demonstrated the effectiveness of our method on a real-world dataset with nine stations. Note that in this paper we detected seismic events on the sliced waveform segments. In future work, we plan to extend our method on continuous seismic waveform data.

Acknowledgement

This work is supported by the Natural Science Foundation of China (Grant No.61976001 and 61876184), and the Natural Science Foundation of Anhui Province (1908085MF219). The authors would like to thank Data Management Centre of China National Seismic Network at Institute of Geophysics, China Earthquake Administration and Northern California Earthquake Data Center (NCEDC) for providing waveform data for this study.

References

[1] R. V. Allen, "Automatic earthquake recognition and timing from single traces," *Bulletin of the Seismological Society of America*, vol. 68, no. 5, pp. 1521–1532, 1978.

[2] M. Withers, R. Aster, C. Young, J. Beiriger, M. Harris, S. Moore, and J. Trujillo, "A comparison of select trigger algorithms for automated global seismic phase and event detection," *Bulletin of the Seismological Society of America*, vol. 88, no. 1, pp. 95–106, 1998.

[3] J. Zheng, J. Lu, S. Peng, and T. Jiang, "An automatic microseismic or acoustic emission arrival identification scheme with deep recurrent neural networks," *Geophysical Journal International*, vol. 212, no. 2, pp. 1389–1397, 2017.

[4] T. Perol, M. Gharbi, and M. Denolle, "Convolutional neural network for earthquake detection and location," *Science Advances*, vol. 4, no. 2, p. e1700578, 2018.

[5] Y. Wu, Y. Lin, Z. Zhou, D. C. Bolton, J. Liu, and P. Johnson, "Deepdetect: A cascaded region-based densely connected network for seismic event detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 62–75, 2019.

[6] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep cnn-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2018.

[8] L. Carnimeo, "A cnn-based vision system for pattern recognition in mobile robots," in *Proceedings of the 15th IEEE European Conference on Circuit Theory & Design, Espoo, Finland*, 2001.

[9] Y. Yu, J. Lin, L. Zhang, G. Liu, J. Hu, Y. Tan, and H. Zhang, "Identification of seismic wave first arrivals from earthquake records via deep learning," in *Proceedings of 11th International Conference on Knowledge Science, Engineering and Management*, 2018, pp. 274–282.

[10] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[11] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 109–117.

[12] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in neural information processing systems*, 2007, pp. 41–48.

[13] G. Obozinski, B. Taskar, and M. Jordan, "Multi-task feature selection," Statistics Department, UC Berkeley, Tech. Rep., June 2006.

[14] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 457–464.

[15] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, no. Nov, pp. 1817–1853, 2005.

[16] A. Jalali, S. Sanghavi, C. Ruan, and P. K. Ravikumar, "A dirty model for multi-task learning," in *Advances in neural information processing systems*, 2010, pp. 964–972.

[17] J. Zhou, J. Chen, and J. Ye, "Clustered multi-task learning via alternating structure optimization," in *Advances in neural information processing systems*, 2011, pp. 702–710.

[18] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with dirichlet process priors," *Journal of Machine Learning Research*, vol. 8, no. Jan, pp. 35–63, 2007.

[19] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011, pp. 521–528.

[20] Q. Zhou and Q. Zhao, "Flexible clustered multi-task learning by learning representative tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 266–278, 2016.

[21] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[22] K. Singla, D. Can, and S. Narayanan, "A multi-task approach to learning multilingual representations," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 214–220.

[23] L. Xiao, H. Zhang, W. Chen, Y. Wang, and Y. Jin, "Learning what to share: Leaky multi-task network for text classification," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2055–2065.

[24] J. Cao, Y. Li, and Z. Zhang, "Partially shared multi-task convolutional neural network with local constraint for face attribute learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4290–4299.

[25] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2051–2060.

[26] X. Li and J. Huan, "Interactions modeling in multi-task multi-view learning with consistent task diversity," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 853–861.

[27] Z. Yang, K. Merrick, H. Abbass, and L. Jin, "Multi-task deep reinforcement learning for continuous action control," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 3301–3307.

[28] X. Zhang, W. Li, V. Nguyen, F. Zhuang, H. Xiong, and S. Lu, "Label-sensitive task grouping by bayesian nonparametric approach for multi-task multi-label learning," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 3125–3131.

[29] "Ncedc (2014): Northern california earthquake data center. uc berkeley seismological laboratory. dataset. doi:10.7932/ncedc."

[30] J. Akram, "Downhole microseismic monitoring: processing, algorithms and error analysis," Ph.D. dissertation, University of Calgary, 2014.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.