

**UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ**

**Colegio de Ciencias e Ingenierías**

**Deep-Learning for Volcanic Seismic Events Classification**  
**Artículo Académico**

**Rodrigo José Arroyo**

**Ingeniería en Sistemas**

Trabajo de fin de carrera presentado como requisito  
para la obtención del título de  
Ingeniero de Sistemas

Quito, 7 de Mayo de 2020

**UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ**

**Colegio de Ciencias e Ingenierías**

**HOJA DE CALIFICACIÓN  
DE TRABAJO DE FIN DE CARRERA**

**Deep-Learning for Volcanic Seismic Events Classification**

**Rodrigo José Arroyo**

**Nombre del profesor, Título académico**

**Noel Pérez, Phd**

Quito, 7 de Mayo de 2020

## Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante:

---

Nombres y apellidos: Rodrigo José Arroyo

Código: 00116977

Cédula de identidad: 1716181654

Lugar y fecha: Quito, 7 de Mayo de 2020

## **ACLARACIÓN PARA PUBLICACIÓN**

**Nota:** El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETHeses>.

## **UNPUBLISHED DOCUMENT**

**Note:** The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETHeses>.

## RESUMEN

En este trabajo, proponemos un nuevo método para clasificar entre spectrograms Long-Period y Volcano-Tectonic utilizando seis diferentes arquitecturas de conocimiento profundo. El método desarrollado utiliza tres redes neuronales convolucionales llamadas: DCNN1, DCNN2 y DCNN3. De igual manera tres redes neuronales convolucionales son combinadas con redes neuronales recurrentes llamadas: DCNN-RNN1, DCNN-RNN2, y DCNN-RNN3 para maximizar el valor del area bajo la curva (ROCAUC) en un datases de spectrogramas de eventos sísmicos volcánicos. Los modelos DCNN-RNN1, DCNN-RNN2, y DCNN-RNN3 alcanzaron los desempeños más bajos debido a que presentaron overfitting, y esto puede ser a causa de la pequeña cantidad de muestras por clase utilizadas para entrenar estos modelos ta complejos. El modelo DCNN1 fue el mejor comparándolo con las restantes redes neuronales convolucionales. Se obtuvo un valor ROCAUC de 0.98 y un valor de precision de 95%. Aun que estos valores no fueron los valores mas altos por cada métrica, estos no representaron diferencias estadísticas entro otros modelos de mayor complejidad algorítmica. El modelo DCNN1 propuesto demostró desempeño similar o superior comparado con la mayoría de métodos estado del arte en términos de métrica de precision. Por ende puede ser considerado como un excelente modelo para clasificar eventos sísmicos del tipo LP y VT basados en sus imágenes spectrogramas.

Palabras Claves: Clasificación de eventos sísmicos volcánicos, modelos de conocimiento profundo, inteligencia artificial, imágenes spectrogramas.

## ABSTRACT

In this work, we proposed a new method to classify long-period and volcano-tectonic spectrogram images using six different deep learning architectures. The developed method used three deep convolutional neural networks named: DCNN1, DCNN2, and DCNN3. Also, three deep convolutional neural networks combined with deep recurrent neural networks named DCNN-RNN1, DCNN-RNN2, and DCNN-RNN3 to maximize the area under the curve of the receiver operating characteristic scores on a dataset of volcano seismic spectrogram images. The DCNN-RNN1, DCNN-RNN2, and DCNN-RNN3 models reached the worst results due to the overfitting, and this happened due to the small number of samples per class employed to train these complex models. The DCNN1 was the best model comparing with the remaining deep convolutional neural network models. The obtained area under the curve of the receiver operating characteristic score of 0.98 and the accuracy value of 95%. Although these values were not the highest values per metric, they did not represent statistical differences against other models that were more complex algorithmically. The proposed DCNN1 model showed similar or superior performance when compared to the majority of the state of the art methods in terms of the ACC metric. Therefore it can be considered as an excellent model to classify LP and VT seismic events based on their spectrogram images.

**Key Words:** volcanic seismic event classification, deep-learning models, artificial intelligence, spectrogram images.

## TABLA DE CONTENIDO

<b>INTRODUCTION.....</b>	<b>10</b>
<b>MATERIALS AND METHODS.....</b>	<b>13</b>
Spectrogram images dataset .....	13
Deep-learning networks .....	13
Proposed method .....	14
Experimental Setup .....	16
Spectrogram image preprocessing .....	16
Training and test partitions .....	17
Deep architecture configurations .....	17
Validation metrics .....	17
Selection criteria .....	18
<b>RESULTS AND DISCUSSION.....</b>	<b>19</b>
Performance evaluation of the proposed model .....	19
State of the art based comparison.....	22
<b>CONCLUSIONS AND FUTURE WORK.....</b>	<b>24</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>25</b>
<b>REFERENCES.....</b>	<b>26</b>
<b>APPENDIX A.....</b>	<b>32</b>

**TABLE INDEX**

Table #1 Comparison Based on the Acc Between Related Previous Works Available in the Literature and the Best Selected Model Produced in this work.....	22
--	----



**FIGURE INDEX**

Figure # 1 An LP (top row) and VT (bottom row) seismic signals examples.....	11
Figure # 2 Performance of proposed deep learning model.....	19

## INTRODUCTION

Volcanic monitoring systems are essential to detect early signs of volcanic unrest and possible reawakening that can lead to eruptions (Tilling, 1996). Amongst the techniques used by scientists to estimate activity inside a volcano, seismicity is considered as one of the most effective tools for monitoring and forecasting eruptions (Schmincke, 2004).

In this regard, a wide variety of approaches have been used in recent years to address the problem of volcano seismic events classification. Machine learning classifiers (MLC) such as hidden Markov models (HMM) (Benitez et al, 2007), boosting strategies (Venegas et al, 2019), decision trees (DT) (Lara-Cueva et al, 2016), random forest (RF) (Rodgers et al, 2016) (Pérez et al, 2020), Gaussian mixture models (GMM) (Venegas et al, 2019), support vector machine (SVM) methods (Pérez et al, 2020) (Curilem et al, 2014) (Apolloni et al, 2009) (Lara-Cueva et al, 2016), and artificial neural networks (ANN) (Langer et al, 2006) (Curilem et al, 2009) (Scarpetta et al, 2005) (Curilem et al, 2009), have used with time, frequency or scale domain features to differentiate the seismic events.

Recently, in (Pérez et al, 2020), a new descriptor to represent volcano seismic event signals using image processing techniques instead of classic seismic signal processing strategies was proposed. In such a method, the apparent differences in the pattern of spectrogram images of long-period (LP) and volcano-tectonic (VT) seismic events, as illustrated in Fig. 1, are exploited, and features are computed using the intensity shape, and texture statistics from their corresponding gray-scale images.

On the other hand, convolutional neural networks (CNN) are particular ANN architectures that are gaining more attention in image analysis contexts (Shin et al, 2016) (Chauhan et al, 2018). They avoid using intermediate, fully connected layers to employ

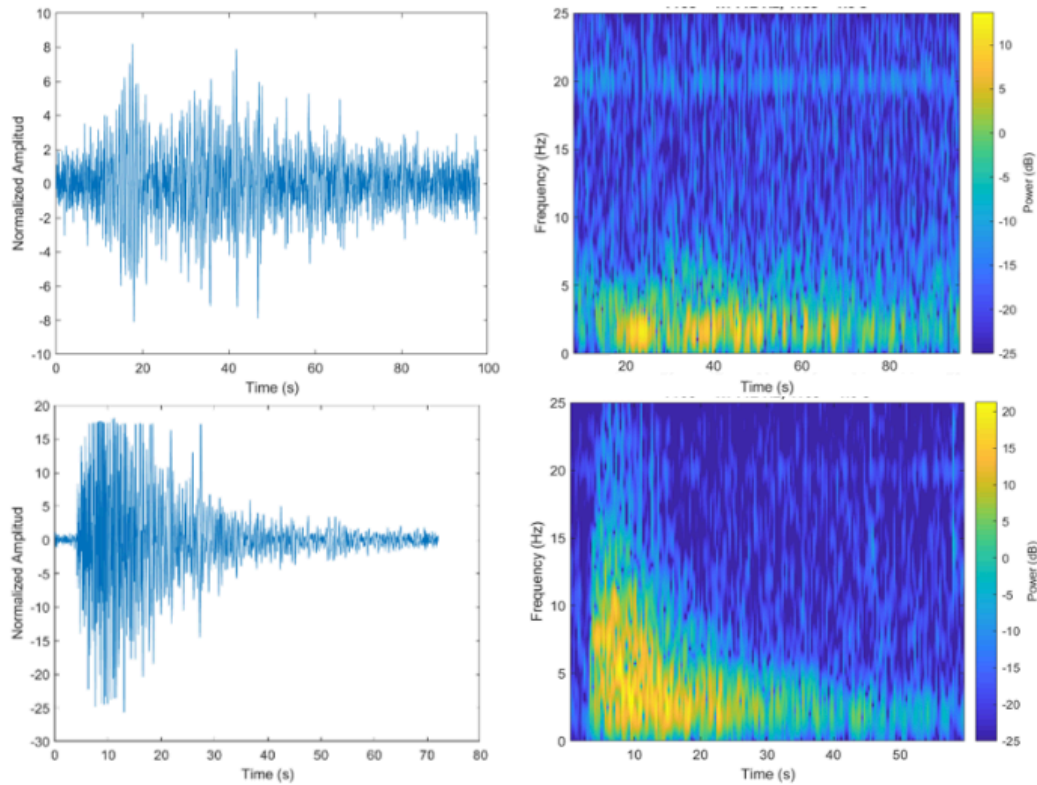


Fig. 1. An LP (top row) and VT (bottom row) seismic signals examples and their respective spectrogram. Taken from [16].

pooling ones and thus optimizing the information pass-through from layer to layer. Lately, there is evidence of using deep CNN models to classify spectrogram images with successful performance (Curilem et al, 2018). Another well known deep learning architecture is the one based on recurrent neural network (RNN) where the information flows sequentially, it is shared between layers and kept as a factor for decision making during the weight calculations (Lecun and Bengio, 2015). The combination of both the CNN and RNN approach is quite possible, as it was demonstrated in (Wang et al, 2016) to classify different objects on individual images. However, the model growing is a vital aspect to take care of. As long as more internal layers are included in the desired model, more complexity is adding as well (Canziani et al, 2016).

Up to date, there are some deep learning-based approaches for object detection (Lecun and Bengio, 2015) (Wang et al, 2016), classification (Yu et al, 2015), and even tracking (Redmon and Farhadi, 2018) . But, there is a lack of research in the context of volcano

seismic event classification based on their spectrogram image. Therefore, in this work, we explore the use of six different deep learning architectures to classify LP and VT spectrogram images. The developed method used single and combined deep CNN and RNN architectures to maximize the Area Under The Curve (AUC ) of the Receiver Operating Characteristics (ROC) curve on a dataset of volcano seismic spectrogram images taken from the Cotopaxi volcano, in Ecuador.

## MATERIALS AND METHODS

### Spectrogram images dataset

This work considered the use of a public dataset (MicSigV1) from the ESeismic repository, which contains several seismic event samples recorded at the Cotopaxi volcano (Pérez et al, 2020). Available at [http://www.igepn.edu.ec/eseismic\\_web\\_site/index.php](http://www.igepn.edu.ec/eseismic_web_site/index.php).

The MicSigV1 has a total of 1187 seismic records from two different seismic stations (VC1 and BREF) installed at the Cotopaxi volcano. This dataset contains samples distributed in five classes: LP, VT, regional (REG), hybrid (HB), and icequakes (ICE). Some examples of seismic events inside this dataset are shown in Fig. 1. Due to the small number of recorded samples from REG, HB, and ICE events, we considered only the LP and VT events classes from the MicSigV1 dataset. The selected samples belong to the same seismic station (BREF) to guarantee the same acquisition protocol and to avoid mixed signals. Therefore, the formed experimental dataset contains a total of 668 spectrogram images distributed in 587 and 81 samples of LP and VT event classes, respectively.

### Deep-learning networks

Deep learning has given the ability to enhance computational models by including multiple layers to process large amounts of data and to improve the learning process. Thus, difficult problems regarding image classification and recognition in the past are presently easier to tackle. The deep CNN and RNN are two special deep learning models (Lecun and Bengio, 2015) (Wang et al, 2016), which are increasing their popularity on sequential data analysis and image labeling respectively.

The deep CNN is a multilayered approach of conventional convolutional neural networks that includes an input layer, a set of hidden layers (which could vary depending on

the network architecture from two to hundreds of layers) and an output layer (fully connected layer). In deep CNN learning, each hidden layer is mainly composed of the CNN architecture core, consisting of at least the convolutional and max-pooling layers. Other configurations extend the basic scheme by adding dropout and flatten layers. This multi-layer structure enables the network to learn different data abstractions while transitioning from layer to layer until reaching the output result (Curilem et al, 2018).

The deep learning RNN is based on the classic feed-forward ANN architecture but it includes an extra working piece called loops in connections. In contrast to the feed-forward ANN, the RNN architecture processes the inputs in a sequential way considering a recurrent hidden state in which the current activation is dependent on the previous step activation. The main drawback is related to long-term sequential data, where the gradients tend to vanish during the training. However, there is a more sophisticated approach to design recurrent units and to avoid vanishing problems known as long short-term memory (Hochreiter and Schmidhuber, 1997). It allows for recurrent units to learn long-term dependencies which are a vital key when developing deep RNN models (Mou et al, 2017).

### **Proposed method**

We adopted both types of neural networks to build the proposed method on the bases of six different deep learning architectures, which are briefly described next:

- DCNN1: this model is based on two convolutional layers with 16 filters and 3x3 kernel size each. Two max-pooling layers (one by each convolutional layer) with a pool size of 4x4, one flatten layer, and a fully connected layer (output) formed by two dense layers (32 and 1 neurons).

- DCNN2: this model contains three convolutional layers with 32, 64, and 128 filters with a kernel size of 3x3 each. Three max-pooling layers (one by each convolutional layer) with a pool size of 6x6 each, one flatten layer, and a fully connected layer (output) composed of three dense layers (32, 32 and 1 neurons).
- DCNN3: this model uses two convolutional layers with 20 filters each, and kernel size of 2x2 and 3x3, respectively. Two max-pooling layers (one by each convolutional layer) with a pool size of 3x3, one flatten layer, and a fully connected layer (output), containing three dense layers (32, 32, and 1 neurons).
- DCNN-RNN1: this model combines two convolutional layers with 20 filters each, and kernel size of 2x2 and 5x5, respectively. Two max-pooling layers (one by each convolutional layer) with a pool size of 3x3, one flatten layer, one dense layer with 32 neurons, one repeat vector layer with a repetition factor of 30 units, two long short-term memory (LSTM) layers the first one with 1024 recurrent units and input shape 30x32, and the second one with 512 recurrent units and a fully connected layer (output), containing three dense layers (32, 32 and 1 neurons).
- DCNN-RNN2: this model uses three convolutional layers with 20 filters each and kernel size of 2x2, 5x5 and 5x5, respectively. Three max-pooling layers (one by each convolutional layer) with a pool size of 3x3, one flatten layer, one dense layer with 32 neurons, one repeat vector layer with a repetition factor of 30 units, two LSTM layers: the first one with 1024 recurrent units and input shape 30x32, and the second one with 512 recurrent units and a fully connected layer (output), composed of three dense layers (32, 32 and 1 neurons).
- DCNN-RNN3: this model uses three convolutional layers with 32 filters each, and with a kernel size of 2x2 each. Three max-pooling layers (one by each convolutional layer) with a

pool size of 3x3, one flatten layer, one dense layer with 32 neurons, one repeat vector layer with a repetition factor of 30 units, two LSTM layers with 512 and 256 recurrent units, an input shape of 30x32, and a fully connected layer (output), containing three dense layers (32, 32 and 1 neurons).

## **Experimental Setup**

### **Spectrogram image preprocessing.**

For all spectrogram images in the MicSigV1 dataset, we downscaled the image dimension to 50% of the original image size. Reducing the number of pixels per image will decrease the volume of information used to feed the learning models. This dataset provides spectrogram images without noise; therefore, the seismic event pattern presented on each image is invariant to the downscaling operation. This operation is widely used in image analysis context with deep learning (Curilem et al, 2018) (Eduardo et al, 2017). Besides, the pixels values of each spectrogram image were normalized using the min-max method (Jain and Bhandare, 2011) to bring them into the range from 0 to 1, thus, avoiding data dispersion.

Additionally, we used a data augmentation technique to increase and balance the number of samples per class. The experimental dataset is distributed in 587 and 81 LP and VT seismic events, respectively. Thus, each spectrogram image was submitted through three operations, such as shearing, scaling, and rotation, as defined in (Mikołajczyk and Grochowski, 2018). Affinity transformations are widely used (Curilem et al, 2018) and allowed us to reach a total of 1108 spectrogram images, which reinforces the models learning process by training them with more samples per class, helping to avoid overfitting.



### **Training and test partitions.**

The stratified 10-fold cross-validation method (López et al, 2006) was applied before the classification step to build disjoint training and test partitions, and to ensure the sample ratio between both types of events for all folds. Thus, individual deep learning models were trained using different training sets, which enable it to learn from different input space representations. Testing on these different sets promotes trustworthy resulting variability in the classification of individual samples.

### **Deep architecture configurations**

For all models, three main hyperparameters were configured to explore the proposed method limits. Thus, the number of iterations (epochs) was set from 50 to 150 with increment step of 50 units; the batch size was tuned to 16,32 and 64 units, and the learning rate used the *adam* optimizer, which is based on an adaptive estimation of lower-order moments (Kingma and Ba, 2014). This optimizer was designed to combine the advantages of the well-known optimizers *AdaGrad* and *RMSProp* (Kingma and Ba, 2014).

### **Validation metrics.**

The classification performance of the proposed method was based on the AUC of ROC curve and Accuracy (ACC) metrics. The statistical comparison among all the classification schemes was conducted using the Wilcoxon statistical test, which ranks the differences in performances of two MLCs (Demšar, 2006). This test provides a fair comparison among them, and therefore a reasonable selection of the best classification model. We used a significance decision value of 5% ( $\alpha = 0.05$ ) for a two-tailed test (Chicken et al, 2013) on all comparisons.

**Selection criteria.**

Since the proposed method explored six deep learning architectures with different hyperparameter configurations, the best model (output) was selected based on the following criteria: (1) the model with the highest AUC score statistically per architecture, (2) if there was a tie rating performance in AUC scores, the one with lower algorithm complexity is preferred, and (3) the highest AUC score statistically among all the selected models by the two previous rules. More than one model per architecture can be selected, if there is not a significant AUC based difference among the classification models. This exception is only valid for the intra architecture analysis. Thus, the proposed method provides only one classification model as output.

The implementation of the proposed method was done with Python programming language version 3.7.4 (Python Core Team, 2019) using *scikit-learn (SKlearn)* (Pedregosa et al, 2011), Keras (Chollet et al, 2015) with *ImageDataGenerator* and TensorFlow backend as well as *sciPy* for statistical analysis (Jones et al, 2001).

## RESULTS AND DISCUSSION

According to the experimental setup section, a total of 54 deep learning models were evaluated using the experimental dataset. The obtained results are summarized next.

### Performance evaluation of the proposed model

The DCNN1 architecture provided seven out of nine classification models using the first selection criterion. This set of classifiers did not present statistical differences in terms of AUC performance when comparing each other. The AUC range of variation was above the 0.95, which is an outstanding classification threshold for any classification problem.

Although the higher AUC score of 0.99 was reached by the model using a batch size of 32 units and 150 epochs (iterations), the remaining models performed similarly statistically.

According to the second selection criterion, the selected classification model in this architecture is the one implementing a batch size of 32 units, 50 epochs, and AUC score of 0.98 (see Appendix A, bold line).

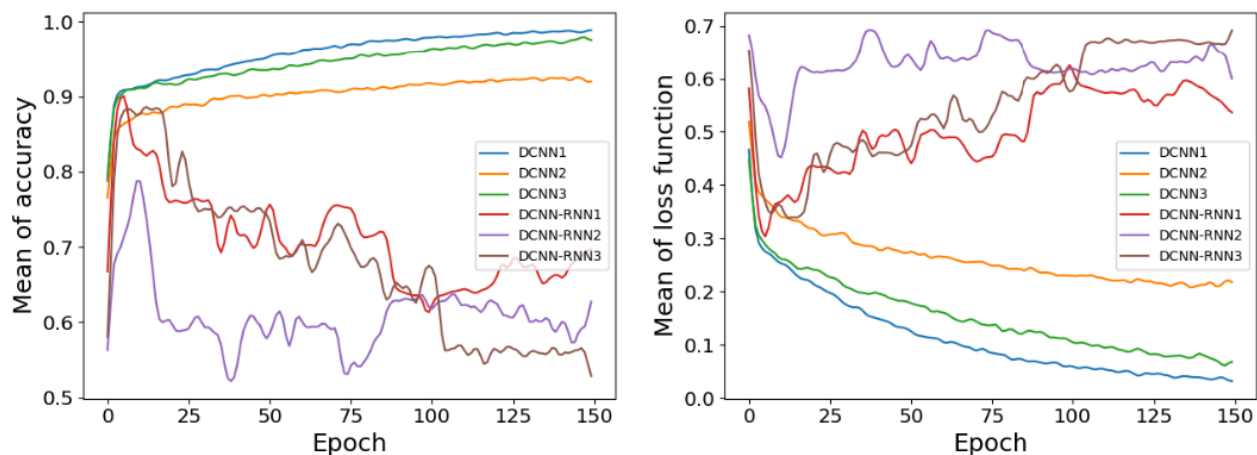


Fig. 2. Performance of proposed deep learning models based on the mean of the accuracy (left) and loss function (right) over ten folds.

Likewise, DCNN2 architecture was able to produce six out of nine classification models that were similar statistically in AUC performances. The range of AUC variation in this set was between 0.71 and 0.79, which are not good enough scores to tackle the problem

at hand. The highest AUC score of 0.79 was reached by the model with 16 units of batch size and 100 epochs. But, the model composed of the same batch size and 50 epochs, which touched an AUC value of 0.71 was selected as the best model of this architecture, taking into consideration the second selection criterion (see Appendix A, bold line).

Similarly, in the DCNN3 architecture, a total of five out of nine classifiers were highlighted as classification models without statistical difference among them. The AUC scores varied from 0.90 to 0.94, which are considered reasonable scores in the context of spectrogram images classifications. The higher AUC value of 0.94 was obtained by the model composed of a batch size of 64 units and 50 epochs. However, there was another model using the same number of epochs respect to the highest model, batch size of 16, and AUC score of 0.91, which was selected as the best model inside this architecture according to the second selection criterion (see Appendix A, bold line).

The combined classification models based on deep CNN and RNN architectures were the worst in terms of AUC performances. The three explored architectures provided AUC scores of 0.50 on all classification models, which means very poor schemes generalization. This effect is extremely linked to the number of samples employed during the models training. Despite using the data augmentation technique and 10-fold cross-validation method on the experimental dataset before feeding the classifiers, they incur in overfitting as it can be seen in Fig. 2. From this figure, it is possible to notice that the mean of the loss function never meets the established learning rate on these models, suggesting the inclusion of more samples in the training process.

The first two selection criteria clearly provided evidence of using classification models only produced by the deep CNN architectures. In opposite to the classification models formed by the combination of the deep CNN and RNN architectures, these three

models overcame the overfitting problem as it is shown in Fig. 2, where they performed over the 90% of the mean of ACC in the validation and the loss values were converging to the learning rate across the defined epochs.

Despite the good performance of these classification models, the best selection model in the DCNN2 based architecture reached an AUC and ACC scores of 0.71 and 90%, respectively. These values are statistically lower ( $p < 0.05$ ) when comparing to the best model selection inside the DCNN3 architecture, which touched AUC and ACC scores of 0.91 and 94%, respectively. The difference in performance is linked to the model complexity inherited from its architecture and the number of samples to train it. The DCNN2 architecture is the most complex among all the developed deep CNN architectures. Thus, it is very reasonable to interpret that this model needs more samples and epochs to learn the feature space properly (see Fig. 2, right plot).

Moreover, the selected classification model using the DCNN1 architecture provided the best performances on both validation metrics. It obtained scores of 0.98 and 95% for the AUC and ACC metrics, respectively. It statistically ( $p < 0.05$ ) overcomes the performance of the remaining models (see Appendix A). This success is related to the DCNN1 architecture, which employed two convolutional layers with only 16 neurons (filters) per layer (lower than the DCNN3 architecture). Thus, it was able to learn from the provided features space satisfactorily (see Fig. 2, right plot). Regarding the third selection criterion, the selected classification model of the DCNN1 architecture constituted the proposed method output and the most appropriate classifier to face the problem of volcano spectrogram image classification.

**TABLE I**  
**COMPARISON BASED ON THE ACC BETWEEN RELATED PREVIOUS WORKS**  
**AVAILABLE IN THE LITERATURE AND THE BEST SELECTED MODEL**  
**PRODUCED IN THIS WORK.**

Method	Number of samples	Computed features	Spectrogram images	ACC* (%)
ANN [5]	914	6	no	97
DT [5]	914	3	no	96
ANN [7]	637	17	no	95
RF [7]	637	17	no	93
linear SVM [11]	914	5	no	97
ANN [15]	1033	8	no	94
HMM [3]	-	5	no	90
GMM [8]	667	2	no	94
CNN [19]	15895		yes	97
SVM [39]	105000	102	yes	92
<b>DCNN1 model</b>	<b>1108</b>		<b>yes</b>	<b>95</b>

ACC - accuracy; \*values rounded to the closest integer

#### State of the art based comparison

Although it is not possible to make a direct statistical comparison against the previously developed state of the art methods because they are based on traditional machine learning models versus deep learning models, and the experimental conditions are different as well. However, we aimed to carry out the comparison based on the ACC scores reported by them, as it is shown in Table II.

The majority of presented machine learning models reached ACC scores ranging from 90 to 97%, being the linear SVM and ANN the models which provided the higher classification performance. From Table II, it is possible to read the proposed method was similar and superior then several state of the art methods in terms of ACC scores. That was possible because deep learning-based approaches are able to learn data abstraction from layer to layer, using different mathematical functions. Meanwhile, machine learning methods, except for

nonlinear models like ANN, attempt to fit the data with a single mathematical function, which limited the learning ability.

On the other hand, the developed method in (Curilem et al, 2018), used a deep CNN model that achieved an ACC score of 97%. This result was superior when compared to the 95% obtained by the proposed method. However, they classified four types of seismic events instead of two, like in this work. Also, they made the training-test validation using an extensive dataset, which provided a decent number of samples during the model learning.

## CONCLUSIONS AND FUTURE WORK

In this work, we explored the use of six different deep learning architectures: three deep CNN (DCNN1, DCNN2, and DCNN3) and three deep CNN combined with deep RNN (DCNN-RNN1, DCNN-RNN2, and DCNN-RNN3) models, for the classification of LP and VT events on a dataset of volcano seismic spectrogram images from the Cotopaxi volcano, and used the AUC classification performance as the primary selection criterion. The deep CNN plus RNN based models reached the worst results due to the overfitting. This effect is due to the small number of samples per class employed to train these complex models. The DCNN1 was the best model when compared with the other deep CNN based models, obtaining AUC and ACC scores of 0.98 and 95%, respectively. Although these values were not the highest values per metric, they did not present statistical difference when compared to other models that were more complex algorithmically. Furthermore, the proposed DCNN1 model showed similar or superior performance when compared to the majority of the state of the art methods in terms of the ACC metric. Therefore it can be considered as an excellent model to classify LP and VT seismic events based on their spectrogram images.

As future work, we plan to increase the number of samples per class to experiment with more complex architectures like the deep CNN+RNN models, which incurred in training over-fitting. We also plan to include other types of seismic events such as tremors and very-long period. Finally, the hyperparameter configurations should also be increased to explore the limits of the implemented models.



## ACKNOWLEDGEMENT

The authors gratefully acknowledge the financial support of Universidad San Francisco de Quito (USFQ) during the development of this project under the Poligrants Program (Grants 12494 and 16916). The authors also thank the Applied Signal Processing and Machine Learning Research Group for providing the computing infrastructure (NVidia DGX workstation) to implement and execute the developed source code. The seismic data used in this study was provided by Instituto Geofísico, EPN.

## REFERENCES

- Benitez, M. C., et al., "Continuous HMM-Based Seismic-Event Classification at Deception Island, Antarctica," in IEEE Transactions on Geoscience and Remote Sensing, vol. 45, no. 1, pp. 138-146, Jan. 2007, doi: 10.1109/TGRS.2006.882264.
- Tilling, R.I. (2013). Hazards and Climatic Impact of Subduction–Zone Volcanism: A Global and Historical Perspective. In Subduction (eds G.E. Bebout, D.W. Scholl, S.H. Kirby and J.P. Platt). doi:[10.1029/GM096p0331](https://doi.org/10.1029/GM096p0331)
- Schmincke HU. (2004) Volcanic Hazards, Volcanic Catastrophes, and Disaster Mitigation. In: Volcanism. Springer, Berlin, Heidelberg
- Venegas P., Pérez N., Benítez D. S., Lara-Cueva R. and Ruiz M., "Building Machine Learning Models for Long-Period and Volcano-Tectonic Event Classification," 2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Valparaiso, Chile, 2019, pp. 1-6, doi: 10.1109/CHILECON47746.2019.8987505.
- Lara-Cueva R., Carrera E. V., Morejon J. F. and Benitez D., "Comparative analysis of automated classifiers applied to volcano event identification," 2016 IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena, 2016, pp. 1-6, doi: 10.1109/ColComCon.2016.7516377.
- Rodgers, Mel & Smith, Patrick & Mather, Tamsin & Pyle, David. (2016). Waveform classification and statistical analysis of seismic precursors to the July 2008 Vulcanian Eruption of Soufrière Hills Volcano, Montserrat.

Pérez N., Venegas P., Benítez D., Lara-Cueva R. and Ruiz M., "A New Volcanic Seismic Signal Descriptor and Its Application to a Data Set From the Cotopaxi Volcano," in IEEE Transactions on Geoscience and Remote Sensing, doi: 10.1109/TGRS.2020.2976896.

Venegas P., Pérez N., Benítez D., Lara-Cueva R. and Ruiz M., "Combining Filter-Based Feature Selection Methods and Gaussian Mixture Model for the Classification of Seismic Events From Cotopaxi Volcano," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 6, pp. 1991-2003, June 2019, doi: 10.1109/JSTARS.2019.2916045.

Curilem, M., Vergara, J., San Martín, C., Fuentealba, G., Cardona, C., Huenupan, F., Chacón, M., Khan, M., Hussein, W. y Becerra Yoma, N. (2014). Pattern recognition applied to seismic signals of the Llaima volcano (Chile): An analysis of the events' features. Disponible en <http://repositorio.uchile.cl/handle/2250/126669>

Giacco, Ferdinando & Esposito, Anna & Scarpetta, Silvia & Giudicepietro, Flora & Marinaro, Maria. (2009). Support Vector Machines and MLP for automatic classification of seismic signals at Stromboli volcano. *Frontiers in Artificial Intelligence and Applications*. 204. 116-123. 10.3233/978-1-60750-072-8-116.

Lara-Cueva R. A., Benítez D. S., Carrera E. V., Ruiz M. and Rojo-Álvarez J. L., "Automatic Recognition of Long Period Events From Volcano Tectonic Earthquakes at Cotopaxi Volcano," in IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 9, pp. 5247-5257, Sept. 2016, doi: 10.1109/TGRS.2016.2559440.

Langer H., Falsaperla S., Powell T., Thompson G., Automatic classification and a-posteriori analysis of seismic event identification at Soufrière Hills volcano, Montserrat, *Journal of Volcanology and Geothermal Research*, Volume 153, Issues 1–2, 2006, Pages 1–10, ISSN 0377-0273, <https://doi.org/10.1016/j.jvolgeores.2005.08.012>. (<http://www.sciencedirect.com/science/article/pii/S0377027305003793>)

Curilem, Millaray & Vergara, Jorge & Fuentealba, Gustavo & Acuña, Gonzalo & Chacon, Max. (2009). Classification of seismic signals at Villarrica volcano (Chile) using neural networks and genetic algorithms. *Journal of Volcanology and Geothermal Research*. 180. 1-8. 10.1016/j.jvolgeores.2008.12.002.

Scarpetta, Silvia & Vulcanologia, Istituto & OV, Sezione & Napoli, & Italia, & Ezin, Eugène & Petrosino, S. & Martini, Mohammad. (2005). Automatic Classification of Seismic Signals at Mt. Vesuvius Volcano, Italy, Using Neural Networks. *Bulletin of the Seismological Society of America*. 95. 185–196. 10.1785/0120030075.

Pérez, N. & Benitez, D. & Grijalva, F. & Lara-Cueva, R. & Ruiz, M. & Aguilar, J. (2020). ESeismic: Towards an Ecuadorian volcano seismic repository. *Journal of Volcanology and Geothermal Research*. 396. 106855. 10.1016/j.jvolgeores.2020.106855.

Kaur, T., Gandhi, T.K. Deep convolutional neural networks with transfer learning for automated brain image classification. *Machine Vision and Applications* 31, 20 (2020). <https://doi.org/10.1007/s00138-020-01069-2>

Chauhan R., Ghanshala K. K. and Joshi R. C., "Convolutional Neural Network (CNN) for Image Detection and Recognition," 2018 First International Conference on Secure

Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 278-282, doi: 10.1109/ICSCCC.2018.8703316.

Curilem, M., Canário J. P., Franco, L., and Rios, R. A. "Using CNN To Classify Spectrograms of Seismic Events From Llaima Volcano (Chile)," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018, pp. 1-8, doi: 10.1109/IJCNN.2018.8489285.

LeCun, Y & Bengio, Y. & Hinton, Geoffrey. (2015). Deep Learning. *Nature*. 521. 436-44. 10.1038/nature14539.

Wang, Jiang & Yang, Yi & Mao, Junhua & Huang, Zhiheng & Huang, Chang & Xu, Wei. (2016). CNN-RNN: A Unified Framework for Multi-label Image Classification. 2285-2294. 10.1109/CVPR.2016.251.

Canziani, A. & Paszke, A. & Culurciello, E.. (2016). An Analysis of Deep Neural Network Models for Practical Applications.

Yu, Q. & Yang, Y. & Liu, F. & S., Yi-Zhe & Xiang, T. & Hospedales, T.. (2016). Sketch-a-Net: A Deep Neural Network that Beats Humans. *International Journal of Computer Vision*. 10.1007/s11263-016-0932-3.

Redmon, Joseph & Farhadi, Ali. (2018). YOLOv3: An Incremental Improvement.

Hochreiter, S. & Schmidhuber, J. (1997). Long Short-term Memory. *Neural computation*. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

- Mou, L. & Ghamisi, P. & Zhu, X. (2017). Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*. PP. 1-17. 10.1109/TGRS.2016.2636241.
- Ribeiro, E. & Uhl, A. & Alonso-Fernandez, F. & Farrugia, R. (2017). Exploring deep learning image super-resolution for iris recognition. 2176-2180. 10.23919/EUSIPCO.2017.8081595.
- Jain, Y. & Bhandare, S. (2011). Min Max Normalization Based Data Perturbation Method for Privacy Protection. *International Journal of Computer & 2*.
- Mikołajczyk, A. & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. 117-122. 10.1109/IIPHDW.2018.8388338.
- López, F. & Garcia Torres, M. & Melian, B. & Moreno-Pérez, J. & Moreno-Vega, J.. (2006). Solving feature subset selection problem by a Parallel Scatter Search. *European Journal of Operational Research*. 169. 477-489. 10.1016/j.ejor.2004.08.010.
- Bao, Y. & Yang, H. & Yang, Z. & Liu, S. & Huang, Y. (2019). Text Steganalysis with Attentional LSTM-CNN.
- Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*. 7. 1-30.
- Hollander, M. & Wolfe, D. & Chicken, E. (2014). *Nonparametric Statistical Methods*. 10.2307/1271371.
- Python Core Team, Python 3.6.9: A dynamic, open source programming

language., Python Software Foundation, 2019. [Online]. Available:

<https://www.python.org/>.

Pedregosa, Fabian & Varoquaux, Gael & Gramfort, Alexandre & Michel, Vincent & Thirion, Bertrand & Grisel, Olivier & Blondel, Mathieu & Prettenhofer, Peter & Weiss, Ron & Dubourg, Vincent & Vanderplas, Jake & Passos, Alexandre & Cournapeau, David & Brucher, Matthieu & Perrot, Matthieu & Duchesnay, Edouard & Louppe, Gilles. (2012). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 12.

Chollet, F. et al., Keras, <https://keras.io>, 2015

Jones, E. & Oliphant, T. & Peterson, P. (2001). SciPy: Open Source Scientific Tools for Python.

Malfante, M. & Dalla Mura, M & Metaxian, J. & Mars, J. & Macedo, O. & Inza, L.. (2018). Machine Learning for Volcano-Seismic Signals: Challenges and Perspectives. IEEE Signal Processing Magazine. 35. 20-30. 10.1109/MSP.2017.2779166.

## APPENDIX A: PERFORMANCE RESULTS OF DEEP LEARNING MODELS SELECTED BY THE FIRST SELECTION CRITERION

Architecture	Conv. layer (f)	Kernel size	Pool size per layer	Dense layer (n)	RV layer (r)	Input shape	LSTM layer (ru)	FC layer (n)	Batch size	Epochs (u)	AUC	Wilcoxon at $\alpha = 0.05$ (p value)	ACC (%)
DCNN1	(16, 16)	$(3 \times 3)$	$(4 \times 4)$	-	-	-	-	(32, 1)	64	100	0.99	0.5	99
									32	100	0.99	0.16	99
									64	100	0.98	0.16	98
									32	150	<u>0.99</u>	-	99
DCNN2	(32, 64, 128)	$(3 \times 3)$	$(6 \times 6)$	-	-	-	-	(32, 32, 1)	64	150	0.98	0.45	99
									16	100	0.79	-	92
									32	100	0.72	0.23	93
									64	100	0.72	0.33	94
DCNN3	(20, 20)	$(2 \times 2)$	$(3 \times 3)$	-	-	-	-	(32, 32, 1)	64	50	0.91	0.26	94
									64	50	<u>0.94</u>	-	97
									32	100	0.92	0.33	98
									64	100	0.91	0.17	99
DCNN-RNN1	(20, 20)	$(2 \times 2)$ $(5 \times 5)$	$(3 \times 3)$	32	30	(30, 32)	(1024, 512)	(32, 32, 1)	16	50	50	-	-
									16	50	50	-	-
									16	50	50	-	-
									16	50	50	-	-
DCNN-RNN2	(20, 20, 20)	$(5 \times 5)$ $(5 \times 5)$	$(3 \times 3)$	32	30	(30, 32)	(1024, 512)	(32, 32, 1)	16	50	50	-	-
									16	50	50	-	-
DCNN-RNN3	(32, 32, 32)	$(2 \times 2)$	$(3 \times 3)$	32	30	(30, 32)	(512, 256)	(32, 32, 1)	16	50	50	-	-

Conv.- convolutional; f- number of filters per layer; n- number of neurons per layer; RV- repeat vector; r- repetition factor; ru- recurrent units per layer; FC- fully; connected; u- units; AUC and ACC - mean of AUC and ACC metrics over ten folds; underlined AUC value is the Wilcoxon test pivot value; ACC - mean of accuracy.