Mathematics and Computer Science
2021; 6(2): 38-44
http://www.sciencepublishinggroup.com/j/mcs
doi: 10.11648/j.mcs.20210602.12
ISSN: 2575-6036 (Print); ISSN: 2575-6028 (Online)

Python Language Training System Based on MFCC, VQ, Variational Coefficient and KNTM Algorithm

Leandro Daniel Lau Alfonso[1, *], Sergio Suarez Guerra[2], Jose Luis Oropeza Rodriguez[2], Roberto Rodriguez Morales[1], Gustavo Asumu Mboro Nchama[3]

[1]Institute of Cybernetics, Mathematics and Physics, Havana, Cuba

[2]Computer Research Center, National Polytechnic Institute, Mexico City, Mexico

[3]Department of Technical Sciences, National University of Equatorial Guinea, Malabo, Equatorial Guinea

Email address:

ssuarez@cic.ipn.mx (S. S. Guerra), leandro@icimaf.cu (L. D. L. Alfonso), joropeza@cic.ipn.mx (J. L. O. Rodríguez), rrm@icimaf.cu (R. R. Morales), becquerrr10@hotmail.com (G. A. M. Nchama)

[*]Corresponding author

To cite this article:

Leandro Daniel Lau Alfonso, Sergio Suarez Guerra, Jose Luis Oropeza Rodriguez, Roberto Rodriguez Morales, Gustavo Asumu Mboro Nchama. Python Language Training System Based on MFCC, VQ, Variational Coefficient and KNTM Algorithm. *Mathematics and Computer Science*. Vol. 6, No. 2, 2021, pp. 38-44. doi: 10.11648/j.mcs.20210602.12

Received: March 31, 2021; **Accepted**: April 26, 2021; **Published**: May 14, 2021

Abstract: This contribution describes the second stage of the creation of a language training system programmed in Python with the aim of application to speech therapy in spanish-speaking countries, starting the study in Cuba. The first stage of this research was carried out in Matlab by analyzing the dynamics of change of the centroids of the codebooks, extracted from words pronounced by a locutor. As second stage, the Variational Coefficient formula is used in order to estimate the percentage of effectiveness with which the announcer performs voice training. A modified approach to programming the variational coefficient is taken into account as a measure of dispersion of a group of vectors. The modification is given by taking the mean of the group of vectors as the vector that represents the phonetic boundaries of the word to be trained. Besides, a novel approach for word recognition is used, based on the K-Nearest Training Matrix (KNTM) algorithm that lays its foundations in the analysis of matrix similarity taken the Frobenius norm as a measure to distinguish similar or non-similar characteristics of a matrix with respect to a database of matrices. To reduce the computational cost of the program and speed up its proper functioning, the training matrices of the database are saved in files with a .tex extension, in this way after training process, the program should only read them and not recalculate them, which significantly reduces the running time of the algorithm.

Keywords: Mel Frequency Cepstral Coefficients, Vector Quantization, Variational Coefficient, Word Recognition

1. Introduction

Speech segmentation and processing have been used in applications such as: computerized databases of speech training systems (especially in voice therapy), voice recognition systems and telecommunication, analysis of vocal dysfunctions, speaker recognition, amplification, echo, and cancellations, among others.

In the study Alani and Deriche [1], a novel approach is proposed for speech segmentation using the wavelet transform, concluding that the results using six wavelet parameters are comparable to those obtained using 16 spectral coefficients of the Mel scale. A Cole-Cole parameter classification with an accuracy of 98.17% for renal calculi types [2] is carried out using k-nearest neighbors (KNN) machine learning algorithm with the 10 nearest neighbors. A gender recognition system with classification model based on SVM and KNN classifier was proposed [3] with help of MATLAB software. The KNN model accuracy was tested for different distance functions such as Minkowski distance, City Block Distance, Euclidean Distance, among others, finding KNN classifier to have higher accuracy than SVM classifier. The use of the variational coefficient to compare measures of evolvability and phenotypic plasticity of traits is discussed in [4], paying special attention to the fact that in many occasions the transformations carried out on the data involved in these calculations are

performed with very little care given to the meaning of the data. An automatic technique of complete speech segmentation is studied [9], with the aim of eliminating the need for manual segmentation of sentences. The phonetic boundaries are stablished through the use of a warping algorithm in dynamic time that requires the use of a posteriori probabilities of each phonetic unit given an acoustic frame. The a posteriori probabilities are calculated by combining the probabilities of acoustic clases (which are obtained from a closing procedure in the feature space) and the conditional probabilities of each acoustic class with respect to each phonetic unit. A novel language model for automatic speech recognition is presented in the research work [11], based on Hidden Markov Model Toolkit (HTK) and compatible with the speech recognition system CMU Sphinx-III. A feature extraction method of SEMG signal based on activated muscle region is proposed in [18] for carrying out hand motion pattern recognition and classification between multi-object groups, using KNN classifier. An Autoregressive model combined with principal component analysis (PCA) and KNN classifier is developed [19] for Atrial Tachycardia, Premature Atrial Contractions and Sinus Arrhythmia in ECG signal data taken at Biomedical Lab, NIT, Jalandhar. As a result, KNN classifier coupled with Burg method has better performance than PCA classifier coupled with YW method. A method for automatic voice data tagging is described in [21]. A new algorithm for the automatic segmentation of the voice based on its phonetic transcription is proposed in [8]. A self-iteration procedure (which does not require training) to find the temporal alignment between vector features and phonetic transcription is used for the proposed method. A model to assign phonemic and phonetic labeling to voice segments is presented in the research work [6]. The model is based on fuzzy algorithms that assign degrees of value to the structured interpretations of syllabic segments extracted from the signal of a spoken sentence, whereby the acoustic interpretation or the phonetic and phonemic characteristics are combined in a hierarchy of rules. A new technique based on an evolutionary algorithm that allows segmenting speech without prior training is proposed in [13]. Recently in the study of Sergio and Jose [17], an alternative solution to the problem of phonetic labeling of words is proposed, based on the monitoring of the dynamics of change of the cepstral vectors associated with the Mel frequency (MFCCs, [5]) that make up the Code Book (LC), extracted from the word to be tagged using the vector quantization algorithm (VQ, [12]). An implementation of the variance fractal dimension algorithm is described [10] as a technique for the analysis of voice waveforms. This technique is also used in the segmentation of speech expressions in sentences, words or even phonemes. The observations were made based on experimental results in digitized voice at 44.1-kilo samples per second, with 16 bits in each sample.

The new in this paper consists on the programmation of a spanish language training system that can recognize the phonetic boundaries of the words and be able to recognize the word the locutor is saying in order to make a virtual speech therapist that does not require internet connection for word recognition.

2. Signal Processing Algorithms

2.1. MFCC, VQ and M1, M2 Filters by Rules for Denoising

For programming the MFCC algorithm in Python is used the rasta.py library founded in GitHub under the URL [20]. Then, the Linde-Buzo-Gray VQ algorithm [12] is implemented taking advantage of the module numpy of Python. In addition, the functions for phonetic labeling and filters by rules M1 and M2 (see [17] for better understanding of M1 and M2) were implemented in Python. The language taken into account for this training system was Mexbet T22 + 6 (see Table 1 of [17]).

2.2. Variational Coefficient

The variational coefficient [4] is a statistical measure of the relative dispersion of a dataset that indicates how large is the standard deviation of this dataset in relation to its mean. It is defined as the quotient of the standard deviation of a sample between its mean. The objective of calculating the Variational Coefficient is to obtain a dispersion measure that indicates in percent how well the speaker has performed his speech training. Let $X=\{x^i\}_{i=1}^N$ a set of N vectors in $\mathbb{R}^n$, the Variational Coefficient Vc is defined as

$$Vc=\sigma/||m||,\ \sigma=(P/N)\ \text{^}\text{½}, \tag{1}$$

$$P=\sum||x^i-m||^2, \tag{2}$$

where σ is the standard deviation of the set X, $||\cdot||$ represents the Euclidean norm in $\mathbb{R}^n$, the symbol ^ means exponentiation and the summation is carried out on the supra index i=1,…,N. The most the vectors of the set X resemble their mean m, then closer to 0 would be the variational coefficient and the less these vectors resemble their mean m, then closer to 1 would be the variational coefficient Vc. Taking advantage of This fact, it is taken as a measure of effectiveness for this software to M* given by

$$M^*=100\cdot(1-Vc), \tag{3}$$

where $\cdot$ represents the usual product of R.

2.3. KNTM Algorithm

The KNTM algorithm (K-Nearest Training Matrix) is an extension to matrices of the KNN (K-Nearest Neighbor) algorithm [2, 3, 18, 19] for vectors, but with some differences in the computational implementation. The main idea of this method consists in obtaining within a set of training matrices, the matrix that is closest to a given one. In the python implementation used, each training matrix has size 14x14. Each row of a single training matrix is formed by the 14 Mel coefficients of the signal voice of one word for training purposes, in this way each training matrix represents 14 different audio signals of the same word. Hence, if we have a matrix, which we want to find within the set of training

matrices, the training matrix that most closely approximates it, we only have to form new matrices obtained from subtracting each training matrix with the given matrix. Let's call each one of the new matrices t-matrices. Then with a defined norm within the space of the t-matrices, we only need to calculate the norm of every t-matrix and we get the t-matrix of minimal norm. In that way, this t-matrix of minimal norm was obtained from the subtraction between the given matrix and one of the training matrices, so finally we select the training matrix from which the t-matrix of minimal norm was obtained. In the programmation was used the Frobenius norm for matrices. The steps to program the KNTM algorithm for word recognition will be listed as follows:

1) To form training matrices of size 14x14, in which every row of a matrix contain the 14 Mel coefficients of the same word, for all the words one desire recognize.
2) Save the training matrices into files with extension. txt. to make a database of audio training.
3) To Form the matrix Mr in which every rows are the same and contain the same 14 Mel coefficients of a word said by the locutor (this matrix represents the word will be recognized by the KNTM algorithm).
4) To calculate the t-matrices by subtracting the Mr matrix from the training matrices.
5) To calculate the Frobenius norm of the t-matrices.
6) Let's make I* the index of the t-matrix of minimal Frobenius norm.
7) To get the training matrix of index I*.

3. Training Process

The aim of this section is to explain the functionality of the training system as well as how the variational coefficient and KNTM algorithm help the locutor to measure an effectivity percentage and to recognize the word spoken, respectively. The interface of the software was programmed with PyQt5 module and is shown in Figure 1.

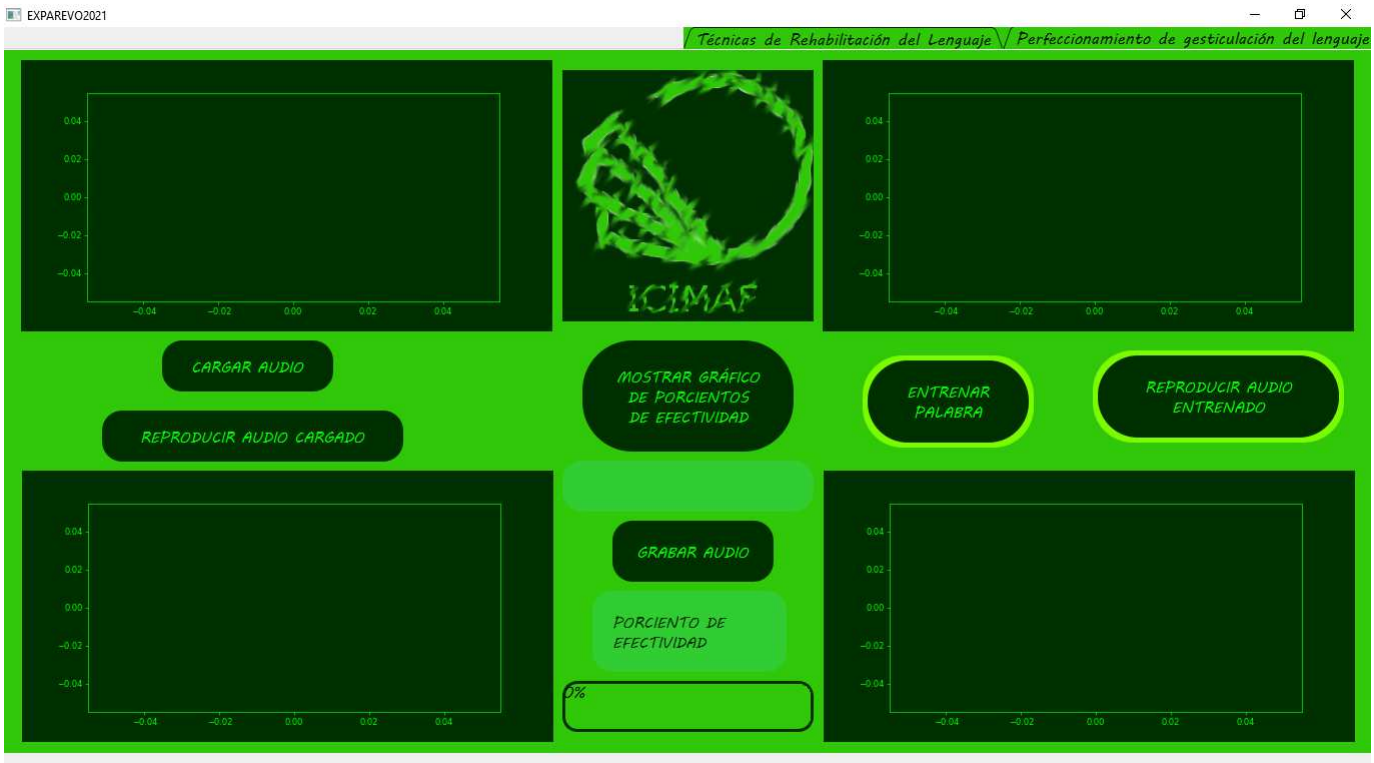

Figure 1. *Interface for the Training System.*

First, the button named Cargar Audio must be tabbed in order to charge an audio file with .wav extension.

Once it is done, a new window is opened with the different audio files for make the training process, as shown in Figure 2. It is selected the audio file ave1.wav and double clicking on it allows to observe the corresponding normalized signal in the left top graphic and the phonetic bounds in the right top graphic for the spanish word ave. Furthermore, it is shown an image related with the spanish meaning of the word ave in the top center, to visualize a representation of the meaning of the word from the speaker part. The phonetic bounds founded are the result of the following steps [17]:

a) Extracting the MFCC coefficients from every portion of 20 ms of the audio file with overlap of 2 ms.
b) A vector of 45 components is formed with each MFCC and its energy, its delta vector and its energy, and its double delta vector and its energy.
c) A matrix is formed taking each row of the matrix as each vector of 45 components.
d) Then the Vector Quantization algorithm is applied to the rows of the matrix to generate the codebook and the codification regions of the Analyzed word.

e) Finally the M1 and M2 filters by rules are applied to this codebook taking into account the dynamical change of the code vectors [17], in order to get the different phonetic segments of the word and to eliminate the noise from the final solution and not getting false phonetics bounds.

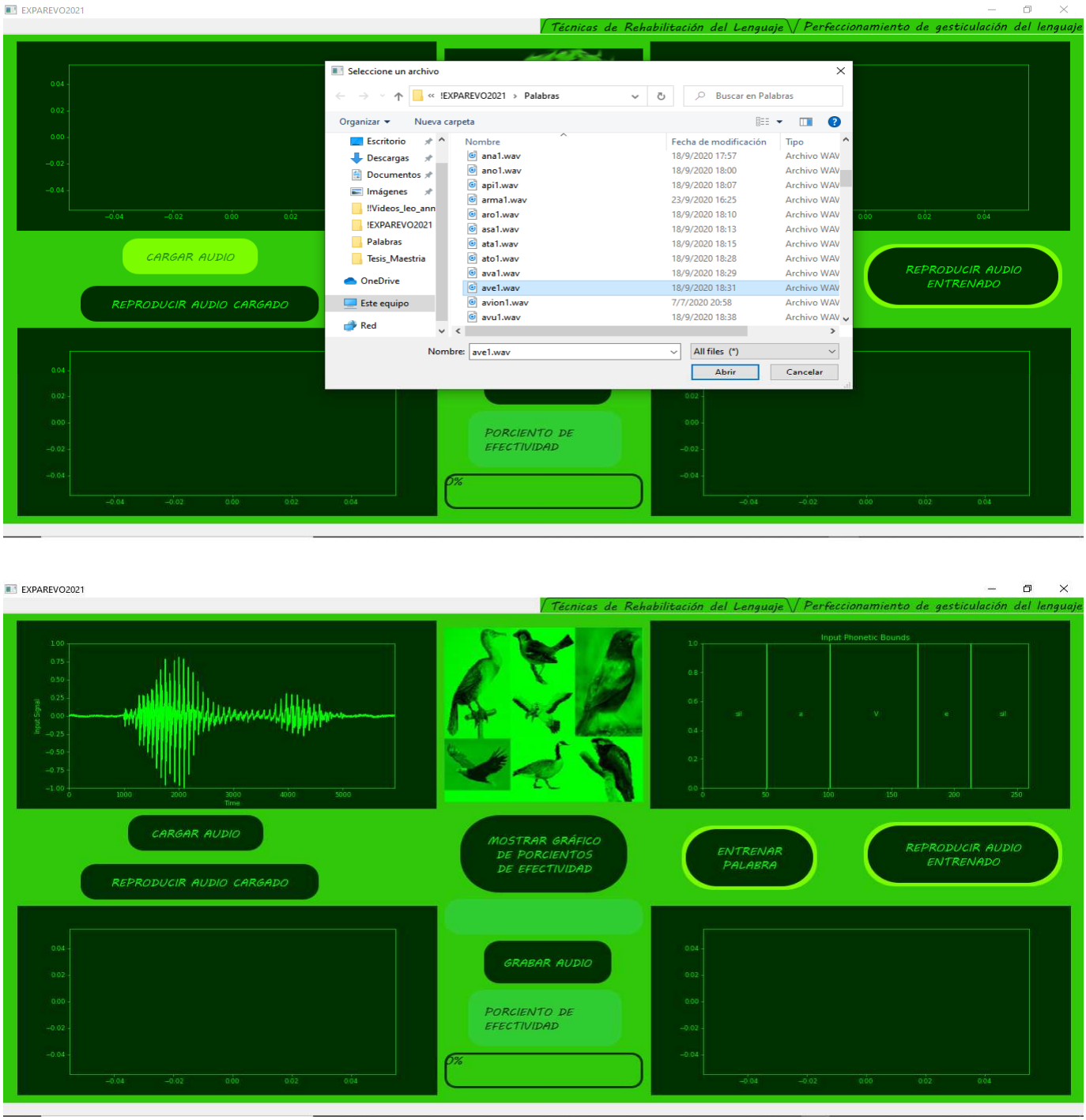

Figure 2. *Charging an audio file.*

The phonetic bounds obtained can be represented in a vector form [17] and from this point on it is used the variational coefficient formulae (1)-(2) in order to measure the effectivity percentage of the speech therapy in the training process.

Once the desired word is charge through audio file .wav, the training process must begin. It begins by tabbing the button Entrenar Palabra in the right side of the interface as shown in Figure 3. With the help of pyttsx3 python engine, once this button is pressed, a spanish voice charged by the interface says: Por favor, repita claramente la palabra, ave. As soon as the voice ends, the locutor must repeat the word and once the locutor finishes the same voice says: Creo que has dicho la palabra, ave in the case the speaker have said this word,

otherwise, for example, if the charge word was ave and the speaker says avion, the voice says: Has dicho la palabra avion, espera por favor, estás diciendo mal la palabra, debes decir ave. The recognition of the spoken word is achieved by means of the KNTM algorithm. It can be performed due to the spoken word is represented by a matrix formed by 14 rows with each rows containing the same 14 Mel coefficients of the word, then the KNTM algorithm is applied to this matrix in comparison with training matrices representing all the words obtained and saved in a dataset of .txt files made by authors. In Figure 3, the normalized voice signal from the word spoken by the locutor is charge in the graphic at the left bottom of the interface and the phonetic bounds encountered in the graphic at the right bottom. Then the vectors of phonetic bounds representing the charged word and the said word are compared by means of formulae (1)-(2), and on this the training process is based. The word to be trained is constantly repeated once it has been loaded, and all the vectors representing its phonetic boundaries are taken as a set of vectors to which it is desired to find its variational coefficient, but taking the mean of these vectors as the vector of phonetic borders that represents the loaded word and not its true mean. This is done with the main objective to obtain a percentage of effectiveness that indicates how good the phonetic bounds are found in the repetitions spoken by the locutor with respect to the phonetic boundaries of the loaded word. The percentage of effectiveness is shown in the form of a progress bar in the lower center of the interface. In Figure 3, a single repetition training of the word ave (bird in english) is shown with an effectiveness of 89% with respect to the charged voice signal.

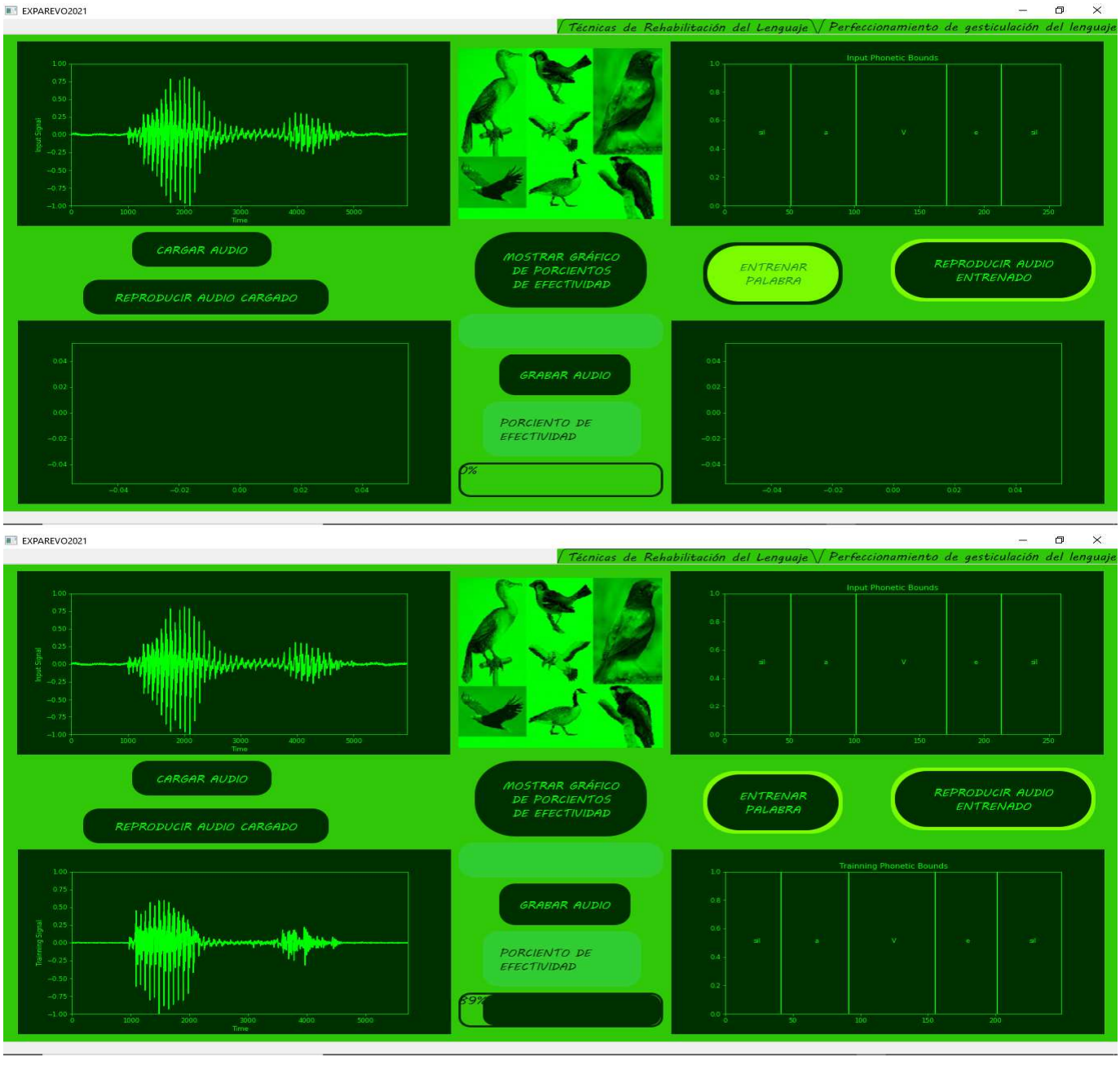

Figure 3. *Training process for audio files.*

4. Conclusion

A language training system for spanish speakers was implemented in python. The system is able to recognize and to show the phonetic bounds of words, through MFCC and VQ algorithms. An effectivity percentage of how well the word is said by the locutor is predicted by the system throughout the computational implementation of the variational coefficient. Besides, recognition of words is achieved through the implementation of the KNTM algorithm. The authors want to propose this system for language physiotherapy in Cuba to people with diction problems, under study in special schools along with speech therapists. As future work, the authors plan to perform voice recognition through hidden Markov models and to find phonetic boundaries of phrases composed of at least two words.

Acknowledgements

The authors are very grateful to Instituto Politécnico Nacional, under the auspices of the project SIP 20181550 y SIP 20195296, as well as the ICIMAF center in Cuba under the auspices of the project: Programa Nacional de Nanociencia y Nanotecnología: Título, "Mejoramiento, segmentación y aprendizaje profundo de nanobioimagenes". Procesamiento paralelo de grandes volúmenes de datos.

Appendix

The python functions for programming the variational coefficient formulation (1)-(2) is shown in Figure 4.

```python
def myVarianza(media, datos):
    N = len(datos)
    desviation = np.zeros(len(media))
    for i in range(N):
        desviation += (np.array(datos[i])-np.array(media)) ** 2
    desviation = desviation / N
    desviation = np.power(desviation, 0.5)
    return desviation
#
def VarCoef(media, datos):
    stndev = myVarianza(media, datos)
    media = np.array(media)
    variationalCoefficient = np.power(np.multiply(stndev,stndev).sum(), 0.5) / np.power(np.multiply(media,media).sum(), 0.5)
    variationalCoefficient = 100 * (1-variationalCoefficient)
    return variationalCoefficient
```

Figure 4. *Variational Coefficient code.*

```python
import numpy as np
#
def vector2matrix(vec):
    N = 14
    exitmatrix = []
    for i in range(N):
        exitmatrix.append(vec)
    exitmatrix = np.matrix(exitmatrix)
    return exitmatrix
#
def matrix2list(mat, tam):
    exitlist = []
    for i in range(tam):
        exitlist.append(mat)
    return exitlist
#
def kNearestMatrix(traininglist, labellist, predictionobject):
    if len(traininglist) != len(predictionobject):
        print('lista de entrenamiento y lista de prediccion de tamannos diferentes')
        return
    else:
        M = len(traininglist)
        restlist = []#para guardar la resta de las matrices
        for h in range(M):
            restlist += [traininglist[h] - predictionobject[h]]
        froblist = []
        for j in range(M):
            froblist += [np.linalg.norm(restlist[j], 'fro')]#guardar las normas frobenius
        froblist = np.array(froblist)
        predindex = np.argmin(froblist)
        predexit = labellist[predindex]
        return predexit
```

Figure 5. *Python Code for KNTM algorithm.*

The python functions for programming the KNTM algorithm is shown in Figure 5.

References

[1] Alani, A., Deriche, M., (1999). A Novel Approach to Speech Segmentation Using the Wavelet Transform. Signal Processing and Its Applications, 1999. ISSPA'99. Proceedings of the Fifth International Symposium on Signal Processing and Its Applications. 1, 127-130.

[2] Banu, S., Cemanur, A., Gökhan, C., Sulayman, J., Tuba, Y., Mehmet, Ç., Bülent, Ö., Ibrahim, A., (2019). Microwave dielectric property based classification of renal calculi: Application of a KNN algorithm. Computers in Biology and Medicine, 112 (2019) 103366.

[3] Bhagyalaxmi, J., Anita, M., Subrat, KM. (2020). Gender Recognition of Speech Signal using KNN and SVM. International Conference on IoT based Control Networks and Intelligent Systems (ICICNIS 2020). Electronic copy available at: https://ssrn.com/abstract=3769786.

[4] Christophe, P., Christoffer, HH, Sigurd, E., Marlène, G., (2020). On the use of the coefficient of variation to quantify and compare trait variation. Evolution Letters, 4-3: 180-188.

[5] Davis, SB, Mermelstein, P. (1980). Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, IEEE Trans. on Acoustic, Speech and Signal Processing, 28 (4): 357-366.

[6] De Mori, R., Laface, P. (1980). Use of Fuzzy Algorithms for Phonetic and Phonemic Labeling of Continuous Speech. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-2 (2): 136-148.

[7] Fant, G. Speech Sounds and Features. The MIT Press, Cambridge, MA, USA, 1973.

[8] Finster, H. (1992). Automatic Speech Segmentation using Neural Network and Phonetic Transcription. Neural Networks, 1992. IJCNN, International Joint Conference on, 4 (4): 734-736.

[9] Gomez, JA, Castro, MJ. (2002). Automatic Segmentation of Speech at the Phonetic Level. En: Structural, Syntactic, and Statistical Pattern Recognition. Lecture Notes in Computer Science, 2396, 883-921.

[10] Grieder W., Kinsner W., Speech Segmentation by Variance Fractal Dimension, Department of Electrical and Computer Engineering and Telecommunications Research Laboratories, University of Manitoba, Winnipeg, Manitoba, Canada R3T 5V6.

[11] Hernandez-Mena, C., Herrera-Camacho, A. (2015). Creating a Grammar-Based Speech Recognition Parser for Mexican Spanish Using HTK, Compatible with CMU Sphinx-III System, International Journal of Electronics and Electrical Engineering, 3 (3): 220-224.

[12] Linde, Y., Buzo, A., Gray RM. (1980). An Algorithm for Vector Quantizer Design. IEEE TRANSACTIONS ON COMMUNICATIONS, COM-28 (1): 84-95.

[13] Milone, DH, Merelo, JJ, Rufiner, HL. (2002). Evolutionary Algorithm for Speech Segmentation. Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, 2, 1115-1120.

[14] Moore, BCJ, Glasberg, BR. (1983). Suggested formulae for calculating auditory-filter bandwidths and excitation patterns, Journal of the Acoustical Society of America, 74 (3): 750-753.

[15] Proakis, JG, Manolakis DG., Digital Signal Processing. Principles, Algorithms and Applications, Third Edition, \copyright 1996 by Prentice-Hall, Inc. Simon \& Schuster/A Viacom Company Upper Saddle River, New Jersey 07458 All rights reserved, ISBN 0-13-394338-9.

[16] Sayood, K. (2012) Vector Quantization. Introduction to data compression (fourth edition) A volume in The Morgan Kaufmann Series in Multimedia Information and Systems, 295-344.

[17] Sergio Suarez Guerra, Jose Luis Oropeza Rodriguez (2020). Automatic Phonetic Labeling at Word Level Using the Dynamics of Changing Codebook Vectors, Computación y Sistemas, 24 (2): 855-868.

[18] Shangchun, L., Gongfa, L., Jiahan, L., Du, J., Guozhang, J., Ying, S., Bo, T., Haoyi, Z., Disi, C., (2020). Multi-object intergroup gesture recognition combined with fusion feature and KNN algorithm, Journal of Intelligent & Fuzzy Systems 38 (2020): 2725-2735.

[19] Varun, G., Monika, M., (2018). KNN and PCA classifier with Autoregressive modelling during different ECG signal interpretation, Procedia Computer Science 125 (2018): 18-24.

[20] Web Site https://github.com/mystlee/rasta_py/blob/master/rasta.py.

[21] Spohrer, JC, Brown, PF, Roth, R. (1982) Automatic Labeling of Speech. Acoustics, Speech and Signal Processing, IEEE International Conference on ICASSP'82, 7, 1641-1644.